LECHT

THE WAVES OF CHANGE

THE WAVES OF CHANGE

A Techno-Economic Analysis of the Data Processing Industry

Charles P. Lecht

ACT

THE WAVES OF CHANGE

A Techno-Economic Analysis of the Data Processing Industry

By Charles P. Lecht

This book presents a global view of the data processing industry, assessing the technological and economic currents by which it has been, is, and (in the author's extensively documented opinion) will be directed in future.

Futhermore, it attempts to account for the impact on developments in the DP field of the psychology of both users and vendors. Patterns of thought and perception, and the ways in which such patterns manifest themselves through judgments affecting the direction taken by technology and product planning strategies, are certainly at least as vital to our understanding of the state of the DP art as are the "purely technical" breakthroughs themselves.

Developments in the realms of hardware and software — both single "events" and "trends" — are addressed at length, as are manufacturer competitive positions, the middle- and long-term impact of the burgeoning micro/mini families, the problems faced by corporate management in assessing the role of hardware/software usage methodologies and the value of specific systems as these apply to their particular needs, and many other subjects of intense interest to all who are affected by DP.

The Waves of Change is a series of wide-ranging essays, closely tied together by a single purpose: to assist the reader in evaluating where we are (and why); and to provoke him to consider the implications of rapid change upon his ability to adapt and survive.

ABOUT THE AUTHOR

Charles Philip Lecht is founder, president and chairman of the board of Advanced Computer Techniques Corp. (ACT), a computer software consulting firm.

Lecht holds a B.S. in mathematics from Seattle University and an M.S. degree, also in mathematics, from Purdue. His involvement in the computer field stretches back to 1951, making him an "old-timer" in a very young industry.

Among his earliest professional activities were programming for IBM's service bureau and for the MIT community's Lincoln Laboratory/MITRE organizations on a variety of scientific and military simulation projects.

From 1960 to 1962, Lecht served in the U.S. Army Ordnance Corps, first as chief of its Programming Division and subsequently of its Mobilization Applications Division.

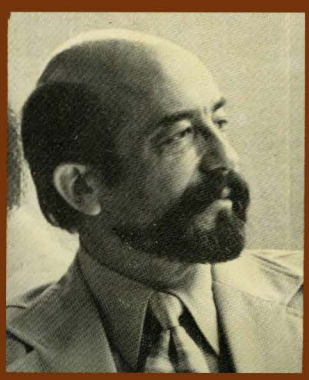
Lecht came to New York City in 1962, where he founded ACT. In the 15 intervening years, the company has grown from a one-man show to an international complex employing some 400 persons and deriving more than 50% of its revenues from operations in Europe, Canada and the Middle East as well as in the U.S.

In addition to building and presiding over ACT, Lecht has found time to hold a number of teaching posts, author five books and innumerable articles and maintain a heavy schedule of speaking engagements in the U.S. and abroad. In addition to *The Waves of Change*, his books include three on computer languages and one on project management.

Lecht is a member of the Young Presidents Organization, The Hudson Institute, the Data Processing Management Association, the Association for Computing Machinery and the New York Academy of Sciences.

In 1976, Lecht was designated by "The Gallagher Presidents' Report" as one of the "10 best businessmen in the U.S.A." representing companies with income below $1 billion. Profiles of Lecht have appeared in the *New Yorker* and *Datamation*, among other publications.

Most recently, Lecht has been requested by the Computer and Business Equipment Manufacturers Association to argue its case against the proposed I/O interface standard before the Brooks Subcommittee.



THE WAVES OF CHANGE

A Techno-Economic Analysis of the Data Processing Industry

To Suzanne

THE WAVES OF CHANGE

A Techno-Economic Analysis of the Data Processing Industry

by Charles P. Lecht
President, Advanced Computer Techniques Corp.

Published by Advanced Computer Techniques Corp.
437 Madison Avenue, New York, N.Y. 10022

ACT, Publishing Division Imprint of
Advanced Computer Techniques Corp.

Domestic Offices:
New York Washington, D.C. Phoenix Tucson

International Offices:
Milan Teheran

Copyright © 1977 by Advanced Computer Techniques Corp.

Library of Congress Catalog Card Number: 77-94890
ISBN: 0-931336-00-7

All rights reserved. No part of this work covered by the copyright hereon may be reproduced or used in any form or be any means — graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems — without permission of the publisher.

Manufactured in the United States of America

Published by Advanced Computer Techniques Corp.
437 Madison Avenue, New York, N.Y. 10022

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

FOREWORD

Hints of the future can often be derived from an historical perspective. Similarly, a sense of history can sometimes help in distinguishing good computer industry analysis and forecasting from bad. It is unfortunate, although not too surprising, that few of us who have a stake in the EDP continuum have accumulated the experience which is gained through protracted involvement in this area: how many readers of this book, for instance, were practicing computer professionals before 1960? All of us are aware of the furiously accelerating rate of technological/political/psychological change within the industry, but how many of us can divorce ourselves from our own particular interests sufficiently to achieve the detachment and objectivity that characterize the historian?

These questions are, of course, partly rhetorical. I am raising them to highlight a need probably already recognized by the reader. . .or why would he be reading this book?

To provide a specific example of what I mean, let us consider the perennial focus on the concept of the product cycle—that is, the five- or six-year period generally defined by use of some particular logic circuit technology. While this cycle certainly defines various industry characteristics, especially of a financial nature, I would suggest that a relatively more "cosmic" view of the industry discloses a cyclic phenomenon with a much longer period, perhaps about 25 years in length.

Computational equipment made its commercial debut prior to 1930 in unique applications, such as census statistics processing. Between 1930 and 1953, more or less, the era of unit record equipment saw the ubiquitous punched card infiltrating most large organizations in a variety of administrative applications.

After 1953, the stored program computer evolved, and while substantial headway has been made in the utilization of this tool, the entire succeeding period may be characterized by hardware functioning as bearer of the bulk of the pricing load (except that sophisticated pricing strategies buried and camouflaged software and support as an assumed cost component).

Now we find ourselves in the midst of a few years of transition which analysts will one day define as having been the threshold of a new cycle, one in which software is king and in which most economic constraints to the long distance transfer of data bits are removed.

As a general rule (i.e., one that is subject to its share of exceptions), I would say that there are some companies that tend to wait passively to watch the future happen, and others that are strongly active in determining its course. In the area of computers, one company has functioned preeminently, albeit not alone, in the latter mode—IBM. We do not wish to minimize the contributions of DEC, Burroughs, Honeywell, Univac, NCR, and others, but in adopting the super macro viewpoint, our attention is irresistibly drawn to the abiding giant.

Its enormous size and resources have allowed IBM such status in the computer marketplace that even significant developments that have had their marketplace origin with other vendors must wait for adoption by IBM before they can be seen as creditable. Thus, for example, while Burroughs is often acknowledged to have been the first computer firm to utilize virtual memory concepts, it was not until IBM embraced them much later that they were universally perceived as valid and marketable. The same point might be made with respect to Univac's role in the effective implementation of real-time and multi-processing concepts. Still IBM was the determining influence, providing the conduit through which those concepts were popularized and disseminated throughout the industry.

My point is that, for better or for worse, IBM is in the position to determine the shape of the "future" at the macro or super-cycle level we are examining here, whether it is originating ideas or merely reacting to ideas introduced by others. From this vaunted level it is able to exercise the kind of power which involves it, ineluctably, in all of our lives.

How does this company come to influence the destiny of our industry, setting aside, for the moment, the contentions of the federal government? Many explanations can be tendered. For one, it has the resources, as we have said, to effect macro level change either as inventor or popularizer. . .intelligence and shrewdness are usually the more persuasive for being underwritten by huge sums of money ($1 billion allocated to R&D in 1976, more than the combined R&D expenditures for all its major competitors in the same year).

But we should really be focusing on economic rather than technological change. IBM is constantly monitoring those evolving economic imperatives which cause large organizations to spend more dollars on information processing, preferably, it hopes, using IBM as prime vendor.

With this in mind, it must occasionally cause or promote volcanic relandscaping in order to thrive. This relandscaping can do double duty in an antagonistic legal environment.

Consider, for instance, what an objective observer might detect as a common boundary condition between the super-cycles which we not so arbitrarily defined. IBM signed consent decrees in both 1932 and 1956, and the next such accommodation may occur by 1980, just as our next super-cycle "officially" begins. Obviously, with a consent decree focusing and reacting to one set of industry conditions, it pays to alter those conditions. And only IBM has sufficient influence-*cum*-resources, not to mention the audacity, to do just that.

That is why so much of this book concerns IBM and the waves of change suggested by current evidence. Since we are at an inflection point regarding industry development, the timing of this book is most apt. But I would suggest readers track the material with a view toward the broad super-cycle whose existence I have proposed here, and perhaps even to the far-out one which should evolve roughly around the turn of the century, because if we are to believe some of the futurological scenarios being advanced, even the alterations that can be identified as being in process may be inadequate to accommodate the explosiveness of information resource growth.

Analyses of what computational environments will facilitate can be mind-boggling. To offer just one example, health care delivery will be revolutionized by 1990, with most large metropolitan areas having implemented vertically-integrated health facilities coordinated by computer. The vertical classifications will include physicians' offices, neighborhood health care centers, hospitals, university medical centers, nursing homes, rehabilitation centers and home health care. A patient entering any part of the system is moved to the level appropriate for treatment of his health problem, and as his condition evolves, either positively or negatively, he may be shifted to a different level. Coordination of the use of facilities is accomplished by computer, which tracks the patient while he is in the system, recording and maintaining his medical and administrative data.

This scenario was one of many produced by a detailed Delphic technique involving 104 "experts" and sponsored by IBM and GUIDE (the IBM equipment-oriented users' association) back in 1971. The study identified 108 specific applications which are virtually nonexistent today but which would likely be implemented at some specified penetration level by the 1980s.

Computer Weekly, a British trade publication, undertook another

Delphic study in 1977, using its broad readership as the expert panel; some of the events which were proposed include:

- Computer-based shopping service available using TV.
- Most software development undertaken by computers, i.e., automatic programming.
- Widespread use of computers for language translation.
- Postal services largely replaced by communications between word processors.
- Widespread use of microprocessors in the home, controlling lights, washing machines, security, etc.
- Commercially available voice controlled typewriters.
- Five megabyte store with random access time less than 100 ms available for less than $160.
- Widespread use of pocket terminals with radio links to computer services.
- Comprehensive information on all citizens stored in a central national computer database.

It is unrealistic to think any of the above will materialize economically without major shifts in industry structure.

Basic references which provide for us the underpinning of our assumptions of the inevitability of change, and at a high rate of speed, are the economic growth theories of Eugene Rostow, and the future shock scenarios of Alvin Toffler. The former develops a long-term theory of economic development applied to nations, which we think is analogous to application development in the information processing age; the three stages of growth in either case are the setting of preconditions, takeoff, and maturity. While we are probably around the convex portion of the presumed Gompertz Curve on a variety of mundane administrative applications, we are in the precondition phase when it comes to the operational side of organizations.

Toffler says persuasively that the rate of change in terms of both invention origination, bringing to market, and dispersal through the economy, is increasing rather than decreasing over time.

I would like to cite one other reference which grasps some of the larger issues at play. In his article "SOS: The Next 11111 Years"

(*Computer*, December 1976), Irwin Tomash perceives the following areas of confrontation which affect our industry:

- The nation-state versus technology.
- The nation-state versus the multinational corporation.
- The nation-state versus its own economy.
- The nation-state versus the computer industry.
- The nation-state versus the telecommunications industry.
- The nation-state versus the semiconductor industry.
- IBM versus the United States.
- IBM versus AT&T.
- IBM versus IBM.
- IBM versus the computer industry.
- IBM versus the user.

To think that a trend-setting (or trend-confirming) organization such as IBM perceives itself as isolated or immune from these various confrontations is giving it too little credit; we would suggest that IBM's strategic plan period (seven years) is implicitly supplemented by an even longer term planning effort, which is quite sensitive to possible future changes in its operating environment. While most executives worry about today rather than tomorrow, firms such as AT&T and IBM can afford the luxury of a further look, which somehow gets factored into the most fundamental decisions.

Thus far, I have suggested that the development of our information processing industry is basically governed by longer term super-cycles, and implied that there is reason to think that such cycles will continue to materialize, given the technological dynamics which exist. It would be naive to think that these technological and other dynamics will not be accompanied by fundamental economic and financial restructuring, which could have wrenching implications, some good, some bad, for users, vendors and investors. That is why I think this book is important. Conventional wisdom as well as vendor propaganda tell us that we are in an *evolutionary* development mode. Nothing could be further from the truth, except to the extent that users' past investments in software will be protected, for a *revolution* is brewing in how the *next* wave of applications will be installed, and the magnitude of the coming application wave will dwarf what has been accomplished historically.

The Waves of Change describes available evidence pointing to the conclusion that we are in transition to a new era. Such evidence should be carefully considered by every student of the computer/communications environment.

Gideon I. Gartner
Oppenheimer & Co., Inc.
New York City, New York
1978

TABLE OF CONTENTS

FOREWORD..... i

PREFACE..... viii

1. COMPUTER INDUSTRY DEVELOPMENTS: A REVIEW..... 1

2. CHANGING HARDWARE/SOFTWARE RELATIONSHIPS..... 19

3. PRODUCTS, SERVICES AND MARKETPLACE STATISTICS..... 35

4. MANUFACTURERS' SPENDING EMPHASIS AND TRENDS..... 49

5. DATA COMMUNICATIONS AND NETWORK TRENDS..... 61

6. DISTRIBUTED PROCESSING..... 75

7. USER BUDGET PRIORITIES — PAST, PRESENT AND FUTURE..... 85

8. FUTURE SYSTEM REQUIREMENTS/EXPECTATIONS..... 95

9. MINI/MICROCOMPUTER IMPACT..... 117

10. SOFTWARE TECHNOLOGY TRENDS..... 133

11. HARDWARE TECHNOLOGY TRENDS..... 165

Figures appear at the end of each Chapter to which they pertain.

PREFACE

This book is about the computer industry. Its purpose is to present an envelope of facts which are necessary (if not alone completely sufficient) to understand the changes taking place within that industry: in its technology, politics, psychology and economics.

Of course, the book's point of view cannot help but reflect the provenance of its constituent data fragments; the reader must be mindful that these have been processed, organized and dispatched by "computer people" who suffer from being caught up in the continuum which they essay, I believe honestly, to describe.

With little further fanfare, therefore, I acknowledge that *Waves of Change,* even at its inception, is necessarily biased because of this. But all bias isn't bad, and some bias can occasionally be more illuminating, in its fashion, than none at all.

Furthermore, I do not pretend that the facts, analyses and conclusions presented in this book are of necessity always correct. This, hopefully, is the case, less because too little research was performed than because of the inherent nature of some computer field ideas.

Our arguments about things seem much more related to differences in philosophy than they are to "simple" facts of engineering. At times, our words seem to serve as mere football signals, invoking abstract plays of great and sometimes unrecognized complexity.

At other times, we acknowledge that the understanding of an idea common to various computer people may be woefully circumscribed, but we forge ahead, using it anyway just to get on with it, hoping for better days.

For example, Chapter 1 contains an illustration purporting to present the numbers of computer systems shipped over the years, as though the universally recognized meaning of "computer system" were the same in the 1950s as it was in the '60s and/or the '70s—or even as though everyone agreed at any point in time on what a "computer system" might be at the same or any other point in time. While it may

seem to some observers (mainly outside the DP field) that our business is defined by an aggregation of concrete objects rather than by abstract concepts, the fact remains that the velocity and fluidity of our technology creates a gray area around even the most fundamental elements of the language we use to describe it.

Is the HP hand-held calculator a computer system? With increasing intelligence being incorporated in previously purely electromechanical devices, e.g., the intelligent disc, the somewhat rigid concept of a computer system that would have been familiar to and unquestioned by the 1950s computer person is the dull and largely remote (in time and practical relevance) antecedent of today's seemingly nonstructured, boundless, dynamic concept.

Similarly, we once spoke of management information systems as though everyone's notion of what they are was identical. Now we have distributed processing with which to contend. And, to our horror, the "computer system" concept (bad enough, as I said earlier) and the "communications system" concept are racing headlong toward each other, perhaps to culminate in some weird, Godzilla-meets-King-Kong, corporate showdown between IBM and Ma Bell.

Anyway I mention all of this because I wish, at the onset, to lay to its long-deserved rest the notion of the possibility of complete certitude in certain key issues in this work. Let the experts and charlatans sustain it with their verbal respirators, if they must. If a concept bearing on an issue under discussion seems vague to the reader, it is probably equally vague to me. Hopefully, we will always be basically "in synch," at least to the extent that we can agree to be perplexed together.

If facts are the message, organization might be considered their envelope. But points of view — historical, conspiratorial, etc. — require participation of both author and reader through a melding of their predispositions to see things in a certain way with respect to other things. Thus, when an opinion or analysis is given, it derives from the way I and/or other computer people see it. If any one conclusion is unpleasant to one reader or another, be assured that the provocation is unintentional.

Finally, I am indebted to many persons whose encouragement and support helped to make *The Waves of Change* possible. Foremost among these are R. T. Fertig, Vice-President of ACT Corporation's Technology-Analysis Group (TAG), whose extensive knowledge of computer industry measures was vital to this effort; and Carl Menkel, ACT's scholarly Manager of Corporate Communications, whose unparalleled editorial skills were always offered with patience and grace.

E. Drake Lundell Jr., *Computerworld's* top Editor, a man unafraid of

innovation, took a "chance" in publishing an excerpted but relatively complete, serialized version of *The Waves of Change* in his widely read newspaper. For this, I owe Drake my most personal gratitude. The assistance of Cheryl Gelb, *Computerworld*'s Chief Copy Editor, and that of her talented colleagues, was also invaluable in bringing that paper's version of *Waves* before the public in coherent form.

To both of Patrick J. McGovern's admirable brainchildren, International Data Corporation (IDC) and Computerworld Communications/Inc., I am indeed indebted. Pat's vision in bringing to the public the hard facts of our industry has paid off for everyone concerned with it — including himself.

Gideon I. Gartner, who contributed the Foreword to this book, in some ways inspired its writing with his excellent Oppenheimer reports on the computer industry, and helped it toward completion through realtime feedback on the *Computerworld* version. A word of special thanks is just as surely owed to the various other outside professional sources, both individual and organizational (and far too numerous to list here), to whom I had recourse in attempting to impart additional credibility to my arguments throughout the book. Their cooperation in permitting me to cite their authority in the field was most welcome.

Ralph Stout, Vice-President/ACT, whose ideas are incorporated into Chapter 10 of this book, and Rosemary Donaldson, upon whose mastery of our word processing system I confidently relied throughout the gestation of *Waves,* both deserve mention. Indeed, Ultra-Text™, ACT's HIS Level 6-based word processing/management workstation, the brainchild of ACT's ultra-bright Vice-Presidents, Ed Bright and John Van Noy, should itself be cited here. This "R2D2" secretary/'droid brought its entire intelligence to bear upon the preparation of *Waves* with never a bleep of complaint.

Charles P. Lecht
New York, New York

1. COMPUTER INDUSTRY DEVELOPMENTS: A REVIEW

From at least one perspective, the computer industry may be thought of as driven by economics and politics — to the dismay of technologists. Many inventions will never see the light of day because of self-impact issues (the replacement of an existing, partly or fully depreciated product with a new, lower priced, higher performance, non-depreciated and lower net revenue device or system.)

Many computer-related developments are not funded even though they may represent the "best" user or market solutions or most cost-effective offerings possible; rather, numerous projects are funded and may eventually lead to the introduction of products, because high-level corporate, political forces have a vested interest in the project.

Product development is often the neurotic offspring of the Muse of Political Compromise, on the one hand, and a company's contending right and left wing forces, on the other. Frequently, this gives rise to an environment wherein low risk is more highly prized than innovation or performance, time schedules are more sacred than ultimate product stability or reliability, and the achievement of immediate cost goals is seen as more vital than long-range maintainability or human factors.

Yes, some technological capability and creativity is essential, but not too much. Was IBM's 370 a major architectural and technological improvement over 360? IBM and many other manufacturers appear basically conservative; i.e., they will not take any unnecessary risks in order to create what the user population may really need. Why upset a core memory manufacturing process just because someone invented integrated circuit (IC) memory (or switch from small-scale to large-scale integration)? Does it make good business sense to render tens of thousands of magnetic tape drives obsolete, thereby losing hard-won rental

revenues on already depreciated equipment, just because some genius in the lab invented a non-mechanical, faster, lower cost and more reliable device?

Whatever the situation, no one would argue with the notion that a viable computer industry requires a proper balancing of technology, politics, psychology and finance with the last most often on the critical path.

Evolution of Complexity

The 1950s/early 1960s was a time of small, simple, free-standing independent computers, during which individual users could use the "machine": a one-on-one relationship. Programmers were also operators, and direct testing and debugging from the console was a common practice.

Applications became more complex, and standard interfaces, controls and so-called "higher level languages" (COBOL, FORTRAN, PL/1, etc.) became commonplace in the mid-'60s. Batch processing and multiprogramming methods were employed to optimize the "machine." Operating systems software evolved, and operators and other specialists came between the "end-users" and the "machine." Small computers grew up and became more complex, medium-scale systems.

Technological hardware improvements were introduced at a phenomenal rate in the early '70s. In 1969 many people hoped for a lessening of the pace of technological change of the 1960s, and in the environments that created it and the products which reflected it. As we entered the 1970s, many, if not all, saw the converse take place in both cases. . .much to their dismay. And, in the latter 1960s, very few people, if any, foresaw with reasonable accuracy the evolutionary path of companies and products, and the impact of these and other phenomena on the world technological scene.

Economies of Scale

The cost of computation, memory, and input/output devices decreased at an accelerated rate (see Figures 1-1 and 1-3). The economy-of-scale, or Grosch's Law, factor encouraged consolidation of applications and databases into fewer, large-scale, centralized systems with elaborate multiprogramming and resource-sharing schemes to optimize the use of the "hardware."

By the mid 1970s, "end-users" were far removed from the computer room, and many data processing departments were too busy supporting and maintaining current, but old, applications. Many, bogged down this way, became totally unresponsive to user needs.

New application developments became a bottleneck; maintenance of the old applications absorbed most of the available resources. Complex real-time or transaction-oriented applications evolved, but many data processing departments, burdened with running day-to-day tactical business applications, were unable to support and/or capitalize upon the growing database/data communications trend.

In some cases, data processing organizations seemed to become ends in themselves, to the dismay of the organization which created them.

Full Circle

By 1980, we will have come full circle; i.e., back to the concept of a "computer for every user" — a one-on-one situation. Intelligent terminals and distributed networks of computers will do much to dispel the frustration experienced by end-users confronting the arcane inflexibility of today's complex systems.

Computer system cost/performance (MIPS, cycle times, data rates, etc.) will play a less prominent role; and "people-performance" — efficiency, responsiveness, ease-of-use, maintainability — will advance to the forefront in the computer system selection process. It should, of course, be noted that MIPS, cycle times and the rest can *indirectly* provide *more functions* to improve people-performance, if they allow higher levels of software usage.

End-user (doctor, lawyer, engineer, librarian, student, etc.) languages, problem-oriented languages (POLs) and application generators will be developed such that the need for additional computer specialists can be significantly limited or reduced.

The distributed processing network era will have arrived, and "people-performance" will be its reason for being. The IBM Share/Guide finding of 1976 that computer industry growth is limited most severely by insufficient software productivity and lack of intelligence in the user community will be well on the way to being overcome by packing more intelligence into the computer system and distributing it more effectively within the organization it serves.

Enhanced Cost Effectiveness

During this inflationary period of rising prices and escalating costs, the data processing industry has been a leader in giving customers and end-users higher performance products at *lower prices*. This has enabled other industries to improve the efficiency of their production and service, to reduce waste, and to minimize the need — if not the desire — for increasing their own prices.

In fact, the emphasis upon technological innovation, the commitment to refinement of product concepts and performance which have characterized the data processing industry throughout the past 25 years have engendered *at least* four generations of product families within that relatively modest timeframe.

Furthermore, the concomitant achievement of enhanced levels of cost effectiveness has been so dramatic that, for example, the cost of performing 100,000 computations has fallen from approximately $1.26 in 1952 to roughly five cents in 1970 and about one cent in 1974. By 1977, this incredible improvement in cost performance approached .5 cent per 100,000 computations (see Figure 1-1).

Prerequisites for Survival

Our ability to classify, synthesize, sort and calculate is the source of much of man's knowledge and achievement. The motivation for processing data faster is not just to save time; with adequate processing speed, problems once considered unsolvable become solvable. Imagine, for example, that the Apollo Program's fastest computers operated at the speed of the Univac I — there would have been no landing on the moon.

Information processing has brought about a revolution enabling us to harness the atom, to automate large and growing portions of our financial, industrial, and governmental enterprises, to fly to Mars and to do many otherwise impossible tasks. Information processing now provides us with a powerful tool — and thus with something far more than a rhetorical invitation — to solve many of mankind's other critical and complex environmental, ecological, economic, political, social and industrial problems.

In our industry, the companies that will survive and grow are those that recognize and act upon the imperative to serve the interests of their customers as well as themselves, and to direct the full energy of their professional staffs to the pursuit of this dual aim. Prosperity, not to say survival, in our business will come solely to those well-managed, functionally integrated firms which are perceptive of and organized to respond rapidly to customer requirements.

Corporations such as General Electric, RCA, Xerox and Singer, which had substantial interests in other areas, failed to make the commitment necessary for success in the computer industry. As a result, we find General Electric's computer division merged into Honeywell's operations, RCA's computer division absorbed by Sperry Univac, Singer

partly acquired by International Computers Limited (ICL) and TRW, and Xerox dropping out of the computer business (its customer base being maintained by Honeywell). This represents a consolidation of many computer manufacturing firms into fewer, more viable enterprises*.

Some *possible* future combinations include: Burroughs and ICL, Sperry Univac and NCR or Phillips, Fujitsu/Hitachi and Control Data Corp. (CDC). More acquisitions, mergers or joint ventures are likely in order to achieve the manufacturing economy of scale (assumed to be about 10% market share) required to survive, and reduce marketing support costs.

One of the real possibilities for the late 1970s is that AT&T, which in 1954 made the first transistorized, general purpose digital computer, may actually enter the "data processing" market. AT&T, with 1976 revenues of approximately $33 billion (twice IBM's 1976 revenues), and drawing upon the considerable resources of Bell Labs and Western Electric, could be a very significant entrant in this industry.

However, in the pre-1976 period we can find many firms which, while they in fact had the resources to compete in the computer industry, nevertheless were unwilling to make the research and development or financial investments necessary to participate effectively. They had the opportunity, but indecisiveness, or lack of adventurous spirit in a business where the state-of-the-art is constantly expanding, caused these firms to be left with rapidly obsolescent products supported by ineffectual marketing programs. Poor management and vacillation imposed a penalty for which technical talent, familiar names and impressive financial position alone could not compensate in a fiercely competitive, ever more sophisticated market.

"Thinking" Computers

This past year was the 200th anniversary of the independence of the United States of America: the first "general purpose" computer was installed by the Bureau of Census in 1951. Twenty-five years may not be an enormously long period in the perspective of the slow but ever-accelerating evolution of modern technology (or nationhood), but it is a long time in the life of an individual, corresponding almost exactly to the time he or she requires to reach maturity, enter the job market or reach at least the middle echelons of business and government.

A whole generation of Americans has thus passed through these phases within a computer-oriented environment. The psychological impact has been enormous: we "think" computers.

---

*Some *new* participants should also be noted; e.g., Itel, Amdahl, Texas Instruments, and various microcomputer firms.

The pervasive role of the computer is today an overriding social as well as business fact. If we examine what has happened in the computer industry and where it is going, it must be in terms not only of straightforward technical and marketing considerations (although they are very important), but also in terms of the generally high level of user awareness of the computer's capabilities and limitations, and of the role we can expect computer networks to play in our lives over the next five to ten years.

What has happened overall to the computer industry to date? Certainly it has been an uneven history with an admixture of incredibly positive factors. . .and some countervailing negative ones, both actual and frighteningly implicit, as well.

Measuring the Positive

We can point to many statistics and inventions to illustrate the "case" for the past technological breakthroughs and dramatic uses of systems which were not even conceivable ten to twenty years ago. In looking back, however, we must also ask the question, "Would it have been better if certain companies had not (and did not) dominated the computer and communications industry and, in effect, inhibit growth, in terms not only of technological progress, but also from the standpoint of application usage?"

Let us look first at some of the positive statistics: the number of installed general purpose computers (see Figure 1-2) has grown exponentially from barely 50 systems in 1953 to 175,000 installations in 1976, and the installed population is projected to nearly double by 1980. However, the definition of what we will call a "computer" in 1980 will certainly have changed substantially compared with that which is accepted vis-a-vis the computers of this period.

At the same time, the number of user organizations has also grown exponentially to over 95,000 in 1976, and is estimated to reach about 240,000 in 1985. In other words, there will be *at least* one installation in every U.S. company or corporation with more than 50 employees in only eight years.

Hardware costs, too, have steadily declined from the $3,000,000 purchase price tag of one-of-a-kind, general purpose machines in 1953, to $380,000 (or about 1/8th of the original price) for a representative "medium-scale" system in 1976.

For example, in 1964-65 a machine in the IBM 360/40-to-360/50 class cost in the neighborhood of $700,000 to $800,000. In 1976, the IBM 370/125-2-to-138 class machine cost less than $500,000 for a total system (Figure 1-3).

System main memory capacity per rental dollar has been a key performance indicator and has also declinded substantially. For example (Figure 1-4A), the IBM 360/75 in the mid-'60s offered a capacity roughly 6.2 times that of the 701 of the early '50s. This has increased to 43.5 times with the IBM 370 Models 158 and 168, and is estimated to increase to over 80 times the IBM 701 capacity when the IBM 30XX models are enhanced with higher density MOS chips in 1978. Note that other manufacturers have already reached this stage; e.g., Sperry Univac's 1100/80, Amdahl's V/7, Honeywell's Level 66, etc.

Another measurement of qualitative enhancement is expressed as the number of cubic feet per million characters of memory. In the early '50s the IBM Model 650 would occupy 400 cubic feet *if* one million characters were available. By the early '60s, this figure was down to 100 cubic feet (1401 and 360/30).

By 1971, there had been a further decrease to 8 cubic feet per million characters (in IBM Models 135 and 145, which utilize bipolar storage), and in 1973, storage space requirements dropped all the way to 1/2 cubic foot per megabyte. From 400 cubic feet to 1/2 cubic foot in about twenty years! By mid-1977, 16K chips became available in production volumes, thus further reducing space requirements to about 1/8th the cubic feet per mega-characters of storage (or nearly .063 cubic feet) needed prior to their introduction (Figure 1-4B).

Figure 1-7 shows the impressive improvement in auxiliary storage devices. For example, the very early disc drives offered 100 bits of information per inch of track with 20 tracks of information per inch of disc radius, and rotated at 1,200 revolutions per minute. One drive offered a maximum capacity of 5 million characters.

More recent disc drives such as the IBM 3330-11 or its competitor equivalents have over 4,000 bits of data per inch of track with over 370 tracks per inch of disc radius, and each disc rotates at 2,400 revolutions per minute. The total storage capacity of a single 3330-11 (or equivalent) is 200 million characters, and the still more recent introduction of the IBM 3350 disc drives (and the competitor equivalents) has nearly doubled that capacity.

Figure 1-7 also shows that one may anticipate bubble or charge-coupled devices to completely supplant fixed-head, fast access disc files or drums such as the IBM 2305 before 1980. Moreover, one may expect disc file capacities of approximately two times that of the 3350, or about one billion characters per drive (two spindles) by 1978. The IBM 3850 mass storage system (MSS) is also expected to double in density and capacity next year, and the introduction of smaller MSS versions of the 138 and 148 computers is anticipated.

The total value (if sold) of installed systems (including communications devices and terminals) grew to nearly $60 billion in 1975. The

number of installed terminal units of all types exceeded the one million mark in 1975, and the terminal market is expected to grow in dollar value by 30% to 40% annually from 1977 to 1980. Approximately 60% to 65% of all systems today use some type of communications device and communications lines. By 1980, it is estimated that 83% to 87% of all systems in the world will be communications-oriented. (See Figures 3-4 and 5-2.)

Software Growth

While the installed base and value of computer systems has grown significantly, so have annual total user DP expenditures, to roughly $38 billion in 1977 and a projected $80 billion by 1980 if the current progression continues. Although extrapolations to the 1985 period may be hazardous, if the trend continues without interruption by recessions or decrease in demand, etc., then user expenditures would grow to $139 billion or nearly 7% of the U.S. gross national product in 1985.

However, the projections of hardware cost decline and potential increases in productivity are based on continuous breakthroughs in the state-of-the-art, not only for hardware, but more importantly for *software.* If the existing significant software bottlenecks and productivity problems are not resolved, then the future growth of this industry will be severely constrained and all extrapolations, forecasts and crystal balls will be of little value.

For example, Figure 1-6 shows that in 1959 the average design and programming cost per instruction (in constant dollars) was roughly $4.50. This climbed to over $7.00 in 1970 and exceeded $8.00 in 1976. Not only users, but also manufacturers are affected by this potentially great inhibitor of industry growth — "software bottlenecks."

The "Lock In" Factor

Currently, the five year development cost of an operating system for a full product line (from mini to maxi), offering a full range of automatic functions and supporting a reasonably global user population may reach as high as $100 million, and the cost trend is definitely up. Moreover, annual maintenance and enhancement costs are growing at a phenomenal rate.

One of the consequences of this trend — the ever-rising software development and maintenance cost — is the "lock in" factor, in conjunction with which user applications software investment must also be considered. IBM has recently estimated* that its customers throughout

*IBM Security Analysts Meeting in 1975.

the world have invested more than $50 billion in applications software alone (double this figure for the total market and vendor software suppliers). As a result of this huge investment, non-compatible systems become prohibitive to consider in view of the enormous conversion costs.

Also worth noting is that while users and manufacturers may have invested $100 billion in "old" software, *change* (in the guise of new data security laws, new tax laws, new database/data communications interfaces, etc.) will force much of this software to be thrown away.

We thus pay homage to the old Chinese proverb that cautions us that "he who mounts the tiger may not dismount."

In the 1977 through 1980 period, the "other" manufacturers will have to concentrate more on expanding and migrating their own customer bases instead of trying to capture a share of the IBM base. Other manufacturers are not only affected by the reduction of their potential market resulting from the economic and technical (complexity) lock-in of IBM users, but also by the steady decline of unit hardware costs (not necessarily user prices).

These vendors are thus forced to increase the total buy by selling larger packages or numbers and varieties of devices in a given installation.

Somehow, the other competing manufacturers, e.g., Burroughs, NCR, Honeywell, DEC, DG, Sperry Univac, ICL, etc., must concentrate, more than at present, on increasing the functionality* of future systems so as to obtain a larger share of the total user EDP budget. Working under IBM's (currently leaking) price umbrella for survival, they are faced with an awesome task to do this — Justice Department, beware of winning too much from IBM!

In a market where exploitation of the installed customer base or capture of a share of the IBM base are becoming at least as important as the search for new customers, IBM's power to set "de facto" industry standards takes on added significance. Through its introduction of the IBM System/32, IBM 5100 and Series/1 minicomputer product lines, IBM is launching an attack to tap the relatively unexploited low-end market.

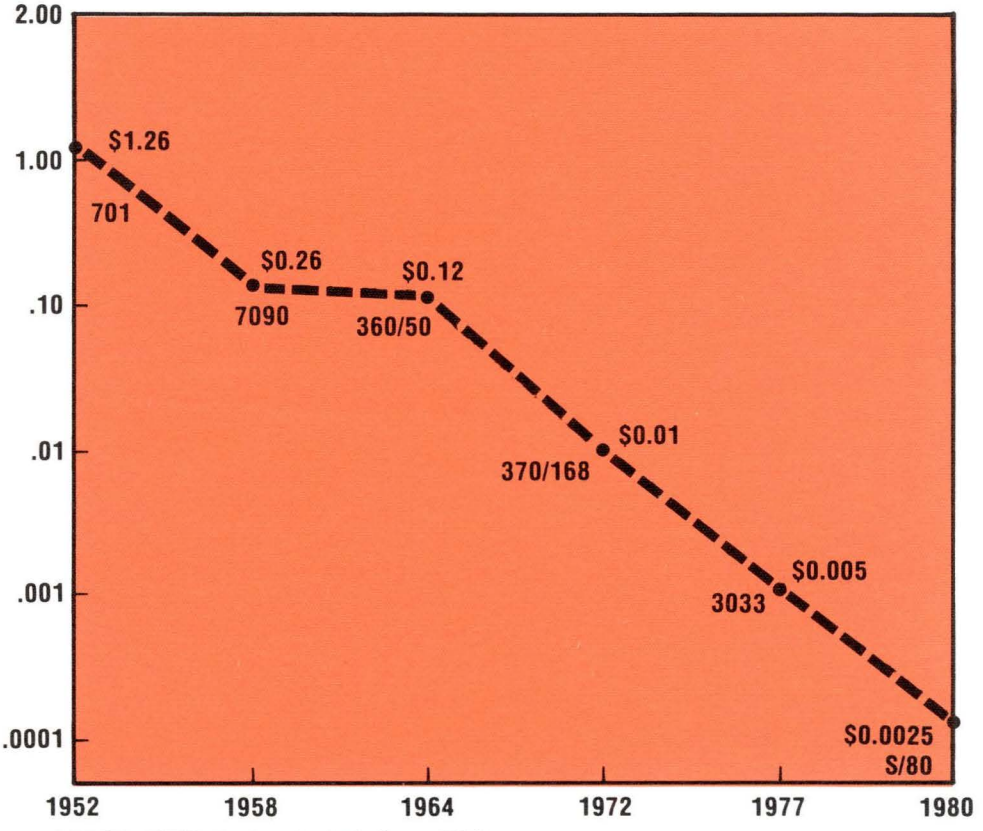
By these steps, IBM has planted the seeds for its future growth, and has gone some distance towards determining the growth impulse and direction of other companies that would compete in the same marketplaces.

---

*The IBM 3850 mass storage system and 3800 laser printers are a case in point. These two subsystems, by reducing the need for operators, librarians, clerks, etc., justify high payments to IBM on a cost-efficiency basis. If labor costs less, the machine can cost more.

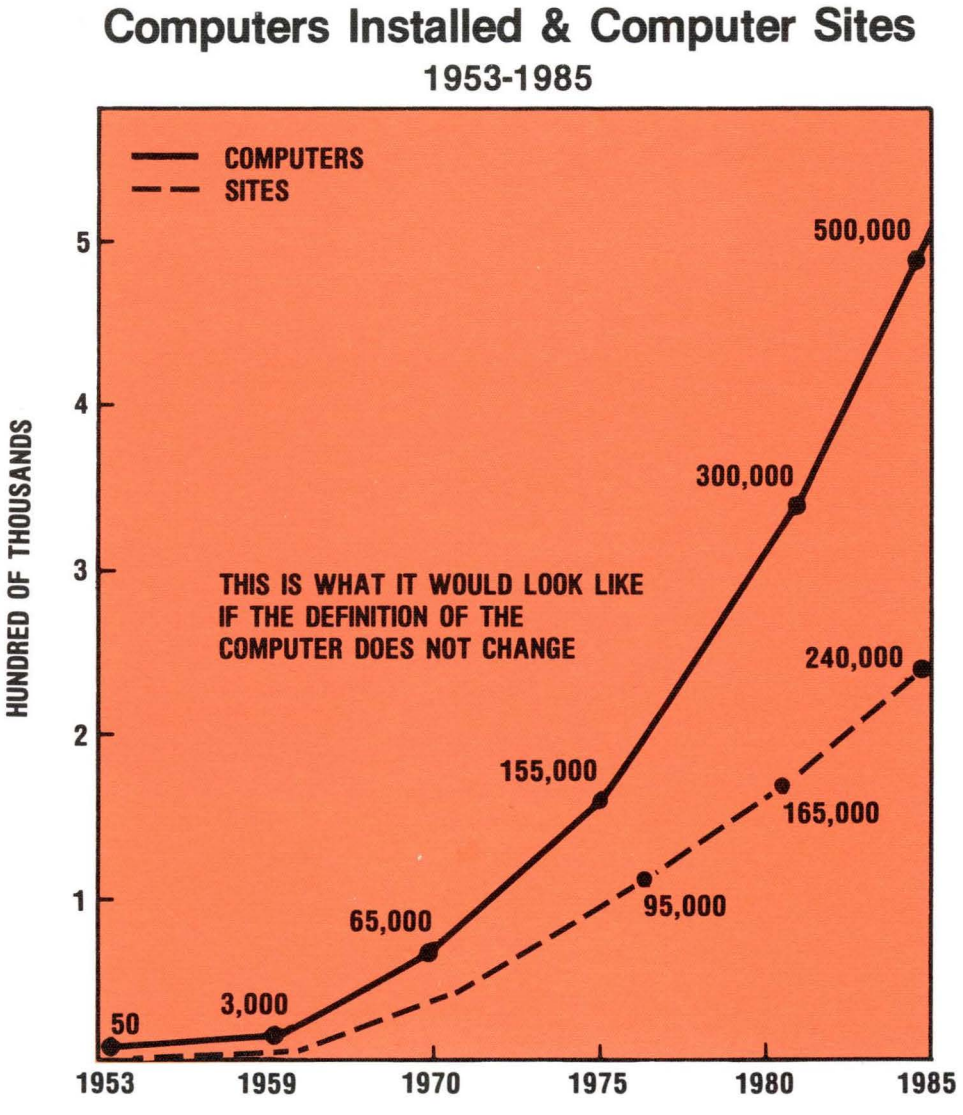
Average Cost Per 100,000 Calculations

1953-1985



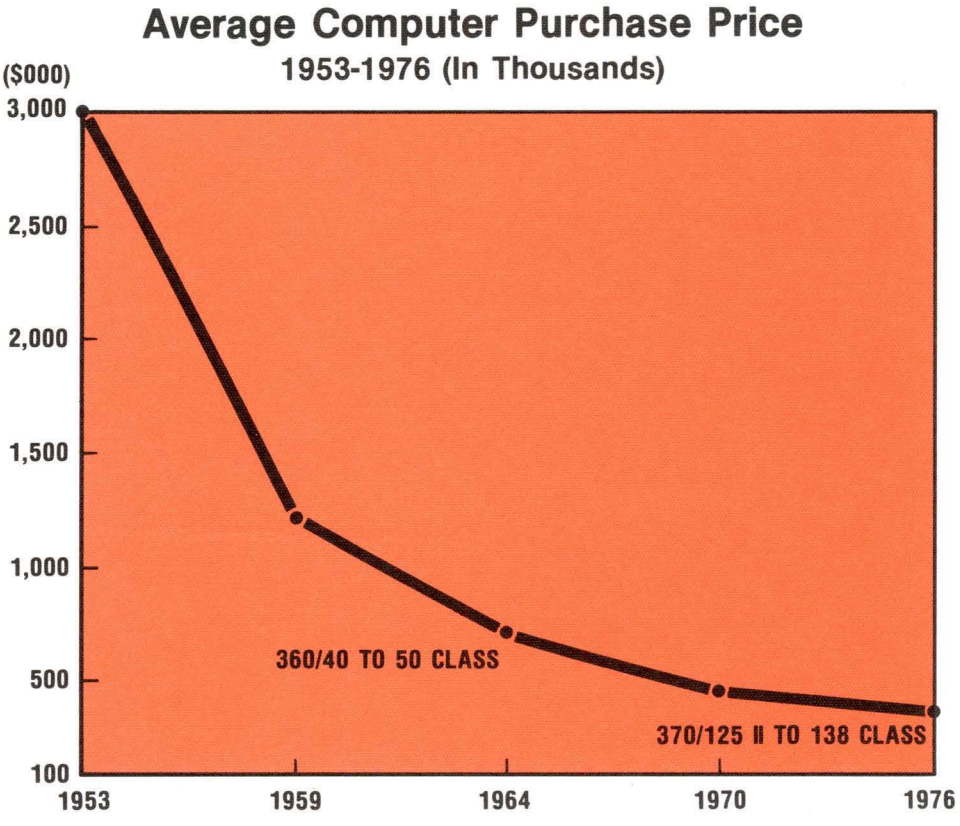
SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 1-1



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 1-2



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 1-3

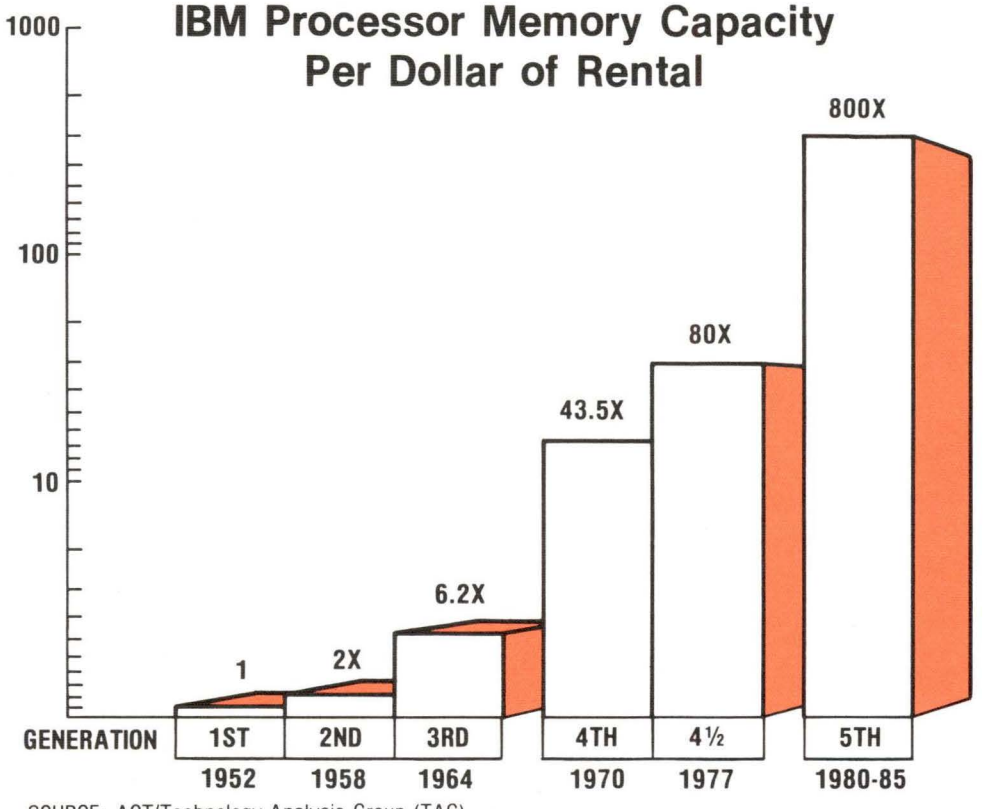
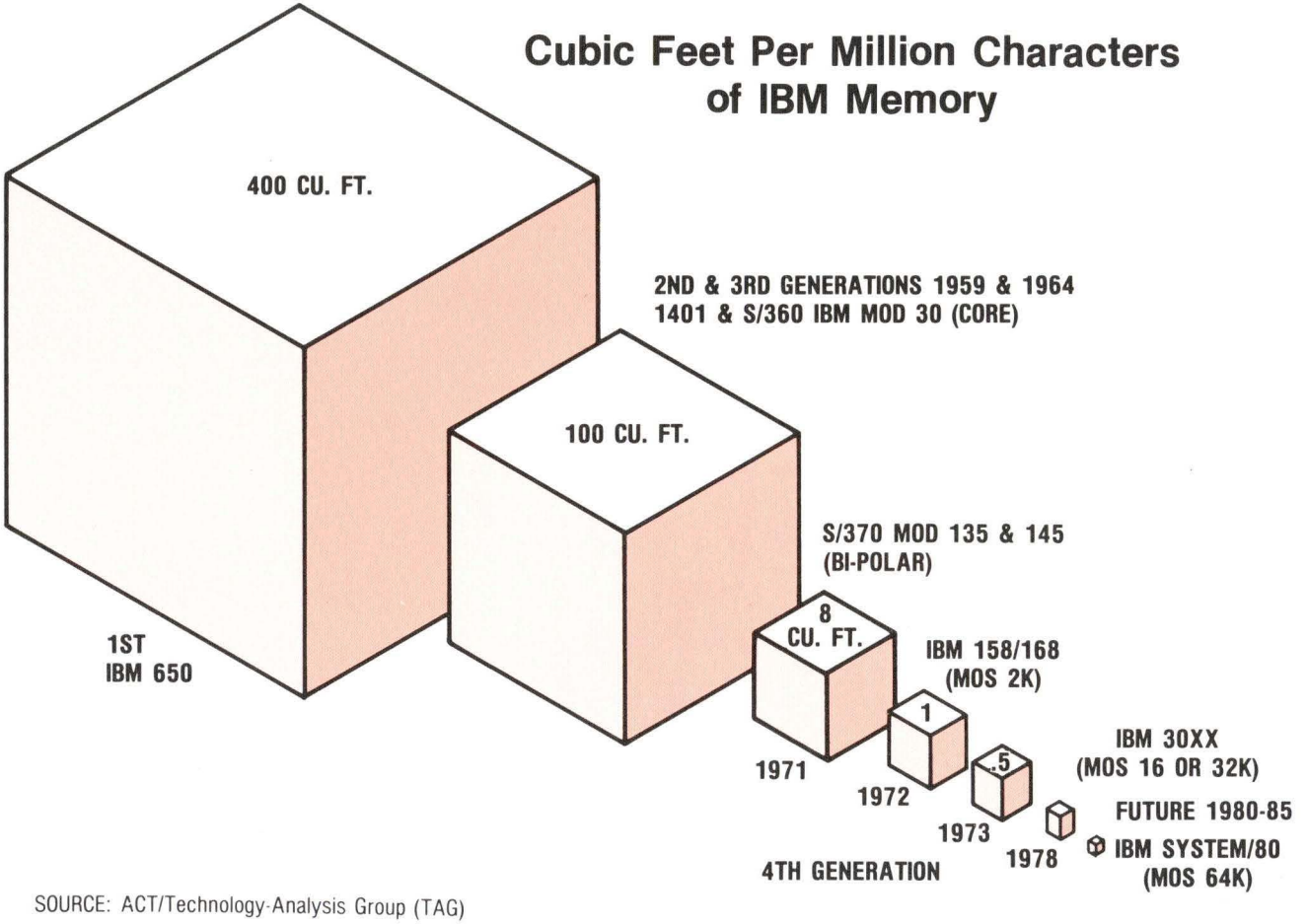
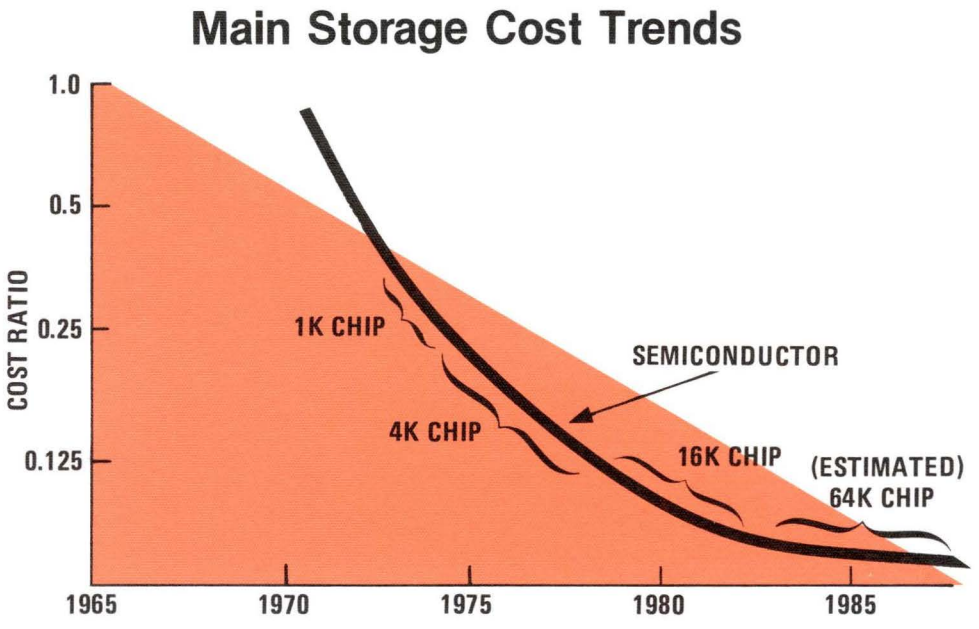

IBM Processor Memory Capacity
Per Dollar of Rental
1000
100
10
1
2X
6.2X
43.5X
80X
800X
GENERATION
1ST
2ND
3RD
4TH
4½
5TH
1952
1958
1964
1970
1977
1980-85
SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 1-4A



SOURCE: ACT/Technology-Analysis Group (TAG)

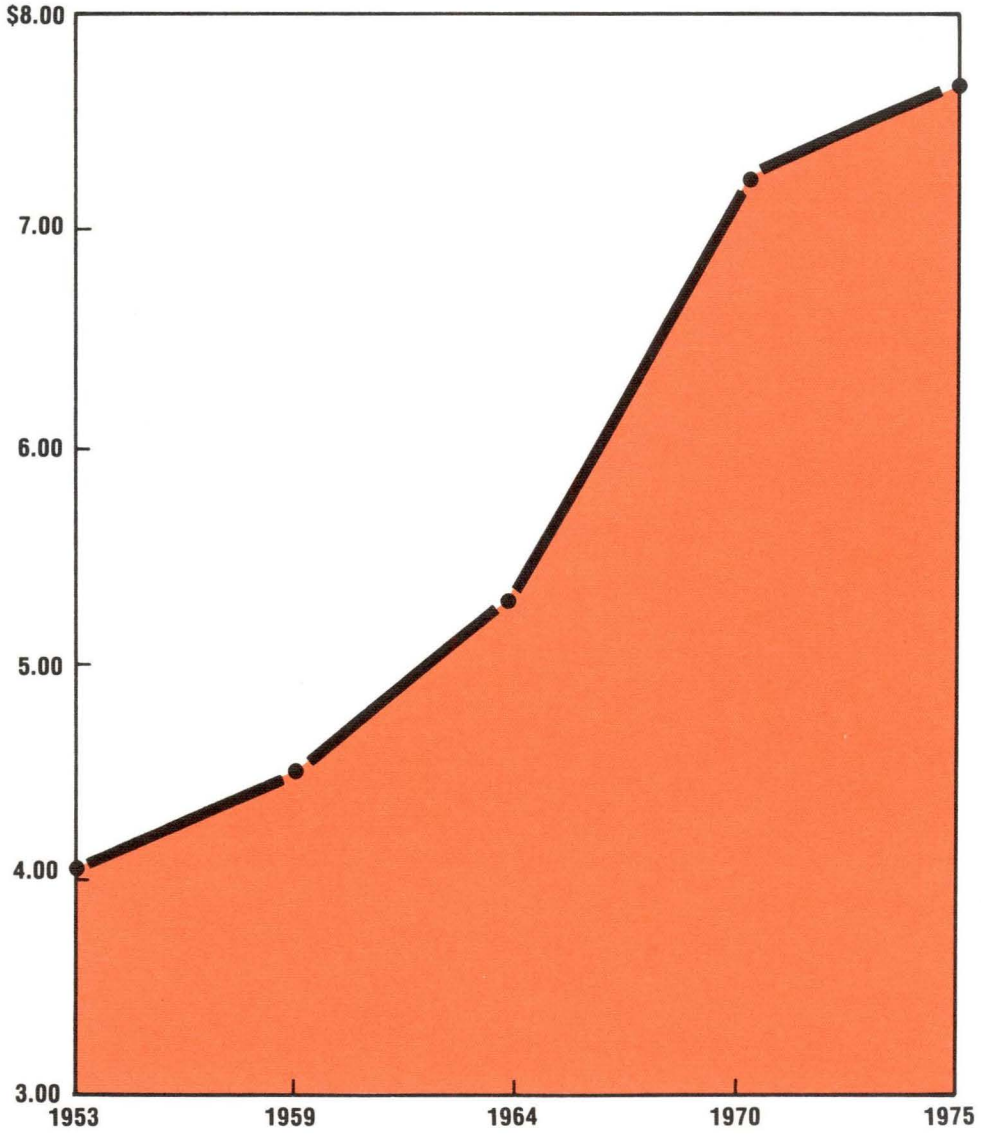
Figure 1-4B



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 1-5

Average Design & Programming Cost Per Instruction
(1959-1975)

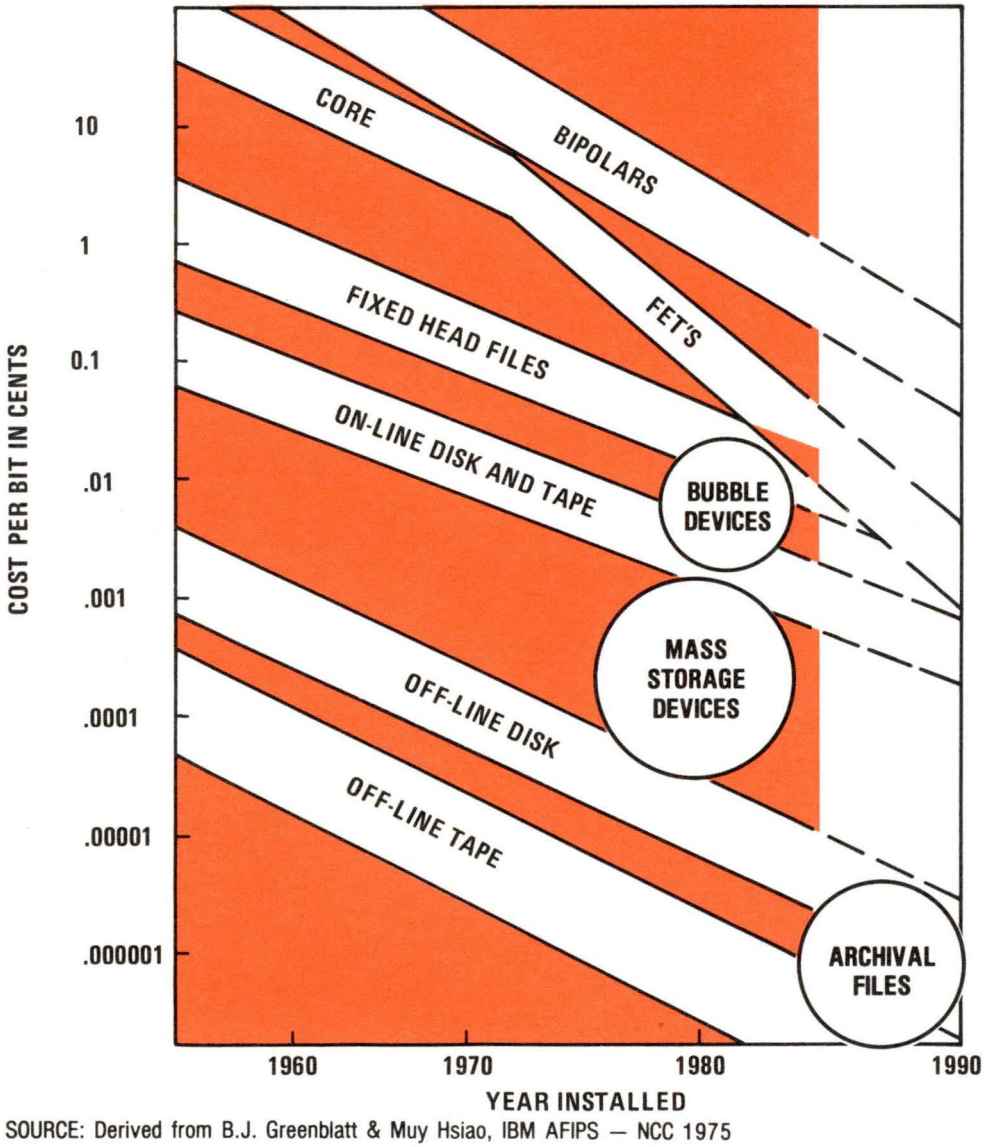


SOURCE: ACT/Technology-Analysis Group (TAG)

SOURCE: A.C.T. Estimates

Figure 1-6

Storage Technologies



SOURCE: Derived from B.J. Greenblatt & Muy Hsiao, IBM AFIPS — NCC 1975

Figure 1-7

2. CHANGING HARDWARE/SOFTWARE RELATIONSHIPS

In the past, many — perhaps most — observers of the computer field tended to overrate new hardware technological developments as a factor in projecting and assessing the future of the computer industry.

Today, however, state-of-the-art techniques in systems software design are emerging as the competitive master key to unlocking new applications developments and installation growth, especially in the medium-to-large-scale area. The major manufacturers' growing recognition of this situation (and its economic ramifications) may explain why the lines between suppliers of computer hardware, software and system services are blurring.

This sort of competitive "overlap" is becoming the rule today not only because of profit opportunities for the vendors (although this must always be assumed to be a key factor); it also reflects the vendors' conscious effort to accommodate changes in priorities of users who are becoming aware, in ever greater numbers, of the importance of software and services in their systems selection decisions.

This shift in priorities, which is giving the edge to vendors whose software makes their hardware easier to use, maintain and operate, is coming about for the following reasons:

- Applications software development continues to be expensive. Even a relatively mundane applications development effort, such as the creation of a typical 20,000-instruction payroll package, currently costs close to $160,000 (up from about $90,000 in 1959, if anyone remembers back that far). In addition, the annual "maintenance" cost for changing applications software modules because of revisions in tax structures, state and local laws and other factors, is a major dollar consideration.
- Basic software costs are steadily outpacing those for hardware. Figure 2-1 illustrates the general trend of software costs as

rising to over 80% of the total hardware/software package over the next ten years; while Figure 2-2 also shows real-life examples of the EDP budgets of two of the nation's important computer customers, the United States Army and the United States Air Force. The latter's software expenditures were projected to exceed $1.25 billion, while hardware expenditures were less than $500 million, nearly a 2.5:1 ratio of software over hardware. Currently, total Defense Department R&D software expenditures are about eight times the hardware costs, according to Dr. Eugene Fubini.

Too much hardware preoccupation on the part of computer systems vendors/users has created a situation not unlike that faced by would-be supersonic aircraft users. However short their actual flight may be, the door-to-door travel time can net out to be longer than they experience with older technology more conveniently located.

Improper allocations of personnel, time and funds to software development in the past sowed the seeds of a malignant contradiction in our creation of an accelerating hardware technology: longer net times to applications implementation with souped-up hardware.

A recent study of future user software requirements by a major user organization (Guide Futures Division, 1975) revealed the following five, not too surprising, facts:

1. Corporate management in most companies wants the ability to predict time, cost and performance of a software project prior to the decision to establish that project.
2. Shorter system and program development cycles are needed to cut down the time required to progress from applications planning to implementation and operation.
3. The vendor must make it easier for the user to make changes in applications packages.
4. Easier access to data is desired, making it possible for management to extract data and reports from the system without going through data processing operations.
5. Users want freedom of output choices, e.g., the ability to use a display or hard copy printer, microfiche, or any other output media.

All five of these points reflect a desire for increased responsiveness from today's data processing environment, and the frustration felt by management as a result of the inflexibility of today's complex systems, and/or the high degree (and cost) of technical talent required to overcome such inflexibility.

This user report concluded "that the current data processing facilities represent a bottleneck. Services are constrained by the limited capacity of the data processing facilities. Present applications systems

are too complex, too inflexible and completely out of control in respect to the purpose for which they were created: to serve the organization. Too many personnel seem to look upon data processing as an *end within itself.*"

In 1965, I prepared as short a statement of the mission of corporate EDP organization as I could. Over ten years later it seems more relevant than ever.

An EDP Charter

The primary mission of EDP is to create the proper *procedural environment* for the continued orderly regeneration of the corporation's *operational environment* as it pertains to:

- The acquisition, purification, storage, processing, retrieval, presentation and dissemination of information for management at all levels; and
- The creation of organic harmony in the integration of man/machine systems into the day-to-day affairs of the corporation.

EDP's primary mission must be supportive of, subservient to, and embedded in the overall management plans, procedures and objectives established by the corporate operating committee. Its role in the corporate environment is unique in that:

- Except in the area of administration of its own affairs, it cannot create a "product" to be delivered to the corporation without dynamic and day-to-day participation by the various corporate sub-organizations which the product is to serve;
- Its success is measured in terms of the proliferation of its capabilities throughout the corporation, and the ultimate loss of its own identity as an independent corporate sub-organization.

Coexisting with the broad objectives outlined above are those bearing on EDP's role in providing guidance in the allocation of funds to be invested in data processing resources, including computer systems, communications systems, etc.

In doing this, it is required to achieve a proper, timely and "in context" capability to ascertain the technical, political, psychological and financial impact of the use of data processing systems in support of the corporation's information-processing requirements, as explicitly directed and/or implicitly inferred by the corporation's operating committee, as well as by its designated authorities.

EDP will prepare and maintain the corporate information systems charter to include long-range objectives, each defined as the outgrowth of a set of short-range accomplishments.

Also, in recognition of the current state-of-the-art in computer systems usage technology, EDP will prepare and maintain its own operational

procedures and standards to serve as an "in-flight guidance system" to ease the transition from systems of the past to those of the future in an atmosphere of realism dedicated to fulfilling the urgent needs of the present.

If one agrees with the charter, the Guide conclusions of 1975 are indeed sad ones which, in fact, might have been reached at any time during the past 25 years. *While not the only reason, software insufficiency alone would have assured this!*

Only recently have the hardware vendors addressed themselves to solving the basic software situation. Language facilities are rapidly changing from the rigidly intellectual symbol manipulation games of the 'fifties and 'sixties to the more natural industrial and/or functional "lingos" of individual disciplinary areas.

Quasi-natural language compilers, database management systems and the like will eliminate the need to maintain the costly activity of the computer programmer as we now know it — COBOLoholics, beware!

By the end of this decade, we anticipate that future systems architecture from IBM, Honeywell, Sperry Univac, Burroughs, NCR and others will be more supportive of EDP's primary mission as given in the charter. Such factors as ease of usage, end-user effectiveness, adaptability and functional capability will be the basis of the success of these systems.

Importance of Language

By ease of usage we mean that their architecture will aid the development of simple, easily used quasi-natural languages. This is of paramount importance to each of the various user classes.

For example, one form of a given language might be more suitable to a doctor, a teacher, a lawyer, an engineer, or a scientist, while still another form of the same language might be appropriate for managers, accountants, security analysts and so forth.

Future user languages will optimize the human resource as well as its hardware counterpart. At the beginning, this will be achieved at the expense of machine cycle time, storage space and other hardware resources. Later, both software and hardware will operate as a unified and synergistic entity, a very worthwhile goal indeed in view of today's trade-offs between the two.

The successful systems vendor must ensure that the "human engineering" question is answered through provision of more human-like language facilities as well as those of a physical nature. Fortunately, some manufacturers are currently working on this.

For example, Honeywell's TDS/IDS and Sperry Univac's TIP/DMS within a COBOL envelope has placed incredible transaction processing/data manipulation power in the hands of its users — eliminating their need to write thousands of lines of code to achieve comparable facilities. Burroughs has been designing its processors with user language facilities in mind.

Both seem to have proven that a few talented people can be more productive in developing advanced applications and/or systems control software with carefully thought out software (in the first case) and hardware (in the second).

Currently, software is costly not only to develop but also to maintain for both the manufacturer and the user. There can be very little market opportunity for new computer systems if users are unable to use them easily, no matter how sophisticated their internal hardware features may be.

In Chapter 1, I made reference to an IBM-sponsored survey of over 160 selected customers in Europe, the U.S. and Japan. The primary objective of this study was to identify the key inhibitors to applications growth in the worldwide computer industry. Looking at the U.S. alone, results showed that *manpower inadequacy* — that is, the lack of sufficient numbers of adequately trained programmers and analysts — was the number one inhibitor. The growing *database and data communications application complexity* problem was second, while *operations complexity* was third.

If current computer industry manpower statistics are to be believed, the only reasonable solution to the dearth-of-talent problem is to pack more "intelligence" into future systems offerings (in the U.S.) or else the market may stagnate and, at best, not grow as rapidly in the late 1970s and early '80s. Manufacturers in or entering the U.S. marketplace: beware! It is evident that those vendors who do not "tune in" to this bottom line fact in the latter part of this decade will fail.*

Advances Lacking

While hardware performance and capacity are growing rapidly because of advances in technology, they are also growing faster than the ability of the current user population to employ them effectively because of lack of advance in software technology.

---

*The "electronics" portion of the computer system price (or cost) is shrinking rapidly. Hence, future technological/cost improvements may have less impact on profit margins.

If current trends continue, by 1980 it is estimated that over 80% of the computer user's total human resources — programmers and systems analysts — will be devoted to the maintenance of old application programs, with less than 20% of these resources available for new applications development. Without new applications development, the U.S. systems market will surely stagnate.

Computer users have changed, both through systematic education and simple, general exposure to three or four generations of hardware and software. Today, even first-time users have considerable awareness of the computer system's capabilities and limitations.

Getting the right machine is relatively easy today; getting the right software with which to do the job effectively is much more difficult. Thus, clients are looking more and more towards software for significant breakthroughs in efficient and effective systems utilization.

Changing Responsibilities

Another indication of how concerned the user is with software ease of usage for new applications development is provided by the shifting pattern in assignment of internal EDP responsibilities. This shift began in the mid-'60s with the formation of project management teams composed of data processing personnel and end-user department personnel to design advanced applications systems.

At first, the project team manager came from the data processing shop. Since 1970, however, he or she has, with increasing frequency, come from the end-user organization.

This trend partly shows that the data processing organization has not been responsive enough to its clients' real needs and that client mistrust (with or without legitimate reasons) is manifest everywhere. It also shows that newly emerging software facilities as well as migration of personnel out of the EDP center and into corporate subentities are lessening the need for non-EDP management to deal with technology per se.

In the past, many user organizations solved software inadequacy problems by acquiring more hardware. It was both easy and tempting for computer manufacturers merely to extend existing architectures through large-scale integration (LSI) and other advanced packaging techniques. But such extension and creation of LSI versions of existing products will not be adequate in the 1980s.

Today, this temptation still exists because one can easily demonstrate that any machine with LSI can be enhanced to achieve three to five times greater CPU performance (measured in terms of millions of

instructions executed per second).* However, yielding to this temptation in today's swiftly changing, increasingly competitive marketplace is dangerous to the manufacturer, not to mention the user.

The policy of extending an existing product line without making radical changes in how it is to be used is, strangely, self-defeating. IBM (we assume) learned this lesson in the early 1970s when it was just starting to install 370s: the recession caused many users to consolidate their multiple medium-scale 360s into one larger 370. The resultant revenue impact on IBM was serious.

Today, no smart manufacturer introduces more powerful systems before users can effectively absorb the extra power.

The reverse is also true; i.e., manufacturers must provide adequate growth paths for their customer bases in terms of increased performance and capacity if they are to retain their clientele. Thus new product announcement policies must seek to achieve a delicate balance between old product payback/old customer retention factors.

Operating System Trends and Unbundling

While we wait breathlessly for something new and different, that is, something which will provide a breakthrough for the user, relieving him of the need to maintain "airplane mechanics" in order to take a routine flight, let's see what's happening. IBM's new Multiple Virtual Storage (MVS) operating system emerges as the key software which, together with Virtual Machine/370 (VM/370), will serve as the bridge for users migrating to System/80 in the latter part of this decade.

Other bridging vehicles include System Network Architecture (SNA), Synchronous Data Link Control (SDLC), the Virtual Storage Access Method (VSAM), and Data Language/1 (DL/1).

With MVS, IBM has developed a modular operating system which will permit future users to select only such software components as are required to meet the needs of their particular installation. It is highly probable that some future version of MVS (Release 4?) will be more fully unbundled on a module basis, so that users may choose specific functions and pay only for the corresponding elements.

This new, "selectable unit" (SU), modular packaging concept was formally announced by IBM in May, 1976, although it can also be seen as an affirmation and continuation of a trend to unbundling and separate

---

*A case in point is the IBM 30XX line, which is merely an extension of the 370 series with some technical improvements and significantly lower prices (about 45% of the pre-March, 1977 price for equivalent 158 and 168 models). This substantial increase in MIPS/$ caused a *temporary* increase in demand for the product.

pricing of system control programming modules which began with IBM's recent offering of a separately priced VM/370 RPQ package for $850 per month.

A more recent development was the introduction in July, 1976, of an SU module for MVS called RACF, which offers significantly enhanced data security at a price of $875 per month. To date over a dozen such SUs have been introduced.

Packaging Technique

A selectable unit is a packaging technique, a software module which represents a performance and/or functional enhancement to the base release of MVS. Formerly, if customers requested a single, new functional or performance enhancement, they had to wait and convert to the next major release of the operating system which, in turn, often contained features many users didn't really need.

With SU, IBM has provided the means for avoiding unnecessarily massive changeovers, since the customer can select only those modules which are truly essential to his needs. Although the SU concept has been announced for MVS, IBM's top-of-the-line operating system, it is inevitable that this approach will spread to other systems as well. In fact, we anticipate that the low-end DOS/VS operating system will eventually be discontinued or put in maintenance mode by IBM as it gradually migrates its customer base to MVS or SVS under VM/370.

The advantages of this flexible software packaging approach are many. For one thing, the new SU scheme crystalizes the interfaces between different functional components of the operating system, which should improve future compatibility and adaptability with System/80. Also, the modular approach facilitates problem isolation, and generally improves the stability of the software from the standpoint of maintenance.

SU packaging gives IBM greater management control over timing of new functional introductions or performance improvements as well.

Moreover, the SU modular approach gives IBM considerably more flexibility in regard to the rate at which it can increase price levels for future extensions or improvements to the operating systems. As additional functions are made available for MVS, SVS, or VM/370, a new system pricing structure can be instituted over the long term, whereby hardware prices will be reduced in near-direct proportion to the software price increases (value-added pricing).

The selectable unit modular approach has implications for IBM's next generation of mainframes (which for lack of a better name we've been calling System/80). The original features of the new system software for "System Q" as it was called in the Telex documents (trial docu-

ments which contained many IBM internal planning papers), included substantial *modularity* and an introduction date of approximately 1976. Furthermore, "System Q" announcements were slated to be gradual, just like the SU concept.

When System/80 is introduced, a large number of new selectable units can be expected, elevating the software pricing level for those users who wish to take advantage of the new hardware capability. Simultaneously, we can expect hardware finally to reflect its true cost by not carrying the burden of software development costs.

Thus, the new generation will be much more competitive at the hardware level than ever before and the *users' costs will shift more and more towards added-value microcode, software and services.* The SU approach allows IBM to phase in a system control pricing strategy on a high controlled basis. There is significant evidence even today that SUs are in fact the prelude to full separate pricing.

Unbundling Leaders

Other manufacturers have really led the way in the areas of operating system unbundling. Control Data Corporation operating systems have been unbundled for over five years.

With the recent introduction of the NCR Criterion line, that company announced that it, too, would unbundle its operating systems: "With software costs constituting an ever greater portion of the cost of the system, it is unacceptable to charge the same proportion of software development cost to each user regardless of the amount of software he is actually using. Now each user will pay for as much as he uses."

Moreover, the Burroughs 80 series (a small-scale system) Master Control Program (MCP) is also unbundled. The SU pricing trend is thus not new.

Why would IBM unbundle its major operating systems at this time? Recent reports from Japan state that Fujitsu has obtained more than 40 orders for the M190 computer and a dozen orders for the M180-II, while Amdahl Corporation has secured at least 90 orders for the 470V/6, including about 30 orders from AT&T.

Furthermore, semiconductor firms in collaboration with other major vendors or leasing companies (e.g., National Semiconductor and Itel) are beginning to market IBM plug-compatible mainframes which use and depend on IBM-developed operating systems software.

Therefore, if IBM has to yield on the relative profitability of its large-scale business because of competition from the plug-compatible mainframers, it will almost certainly place increased value on separately priced systems software. In other words, the user will see his hardware

costs for CPUs, storage, I/O channels and so on decrease sharply over the next several years; but the systems software costs will rise proportionately as the mainframe industry shifts its value-added emphasis from hardware (or firmware) under the pressure of these emerging competitors from Japan and the semiconductor industry.

The end result may be that plug-compatible vendors such as Amdahl, Itel, Fujitsu, Hitachi and CDC may be forced to pay IBM thousands of dollars monthly for selectable units of the operating system used on their equipment (assuming, of course, that they can remain compatible with IBM hardware, microcode and software in the future).

Also, Amdahl's position in particular, in view of its current "piggyback" software mode (piggyback on IBM, that is), may be weakened. Maintaining various versions of various software modules in varying environments may cause its current software support budget to explode.

The IBM selectable unit modular approach may ultimately evolve into a multi-module library offering greater software/microcode optimization or tailoring to suit the user's specific functional and performance requirements. For example, IBM's future systems could be tailored and optimized for particular types of customers (banks, insurance or retail businesses, etc.) and/or applications classes (inventory control, accounting, etc.) via selected, partly microcoded modules.

The SU technique may also make it more difficult for plug-compatible mainframers to unhook or convert IBM systems; i.e., the IBM customer base will be more fragmented, each customer having a different configuration of hardware and SUs which are *proprietary* to IBM.

In addition, this concept, with the proper "microcode lock and key mechanism," may allow IBM to police its unbundling policy to ensure that all users pay their fair share for all systems control programs and program products.

And, finally, during the System/80 era, "pluggable firmware modules" may become available, giving IBM greater flexibility and control over hardware, software usage and user performance and, as noted before, a "functional pricing philosophy" may ultimately unfold.

Data Management Software Trends

A 1976 survey of nearly 600 medium-to-large-scale IBM 370 accounts using database management systems showed that less than 200 were using IBM's new IMS/VS package, while approximately another 100 were using the older IMS 2. Thus, only one-third were using IMS/VS.

Many users seemed to prefer Cincom Systems' Total (120 users). When asked whether current IBM data management software met the

users' near-term needs (that is, two to five years out), more than 50% responded that they viewed the current IBM offering as "inadequate."

At a recent (mid-1976) meeting of IBM users (reported by the Institutional Research Department of Morgan Stanley & Company), IBM's key offering for large-scale database customers was given very low marks by those who had attempted to implement it. Reports regarding performance and lack of flexibility were sufficiently negative that many users are reluctant to consider migrating to IMS and may wait for an entirely new IBM-developed database system in the latter part of this decade.

IBM's failure to develop a satisfactorily performing, easy-to-use database system undoubtedly will cost the company significant drag-along hardware systems sales and, hence, growth.

Long-Term Problems

IMS/VS apparently has substantial long-term problems which do not appear easy to remedy, according to Dr. Weil, Morgan Stanley's computer industry analyst. For example:

1. Some users consider IMS/VS an "architectural monstrosity," as it is based on OS/360 technology which includes much that is now redundant within the Multiple Virtual Storage (MVS) structure.
2. IMS/VS uses its own access facilities and scheduler rather than utilizing those of MVS. This results in rigidities and performance penalties which a number of users find intolerable.
3. The IMS structure is much too limited in the data inquiry or retrieval applications arena at a time when IBM is beginning to consider senior management and end-users as the most likely sources of continued growth in computer capacity (and revenue).
4. IMS is built on a hierarchical concept as opposed to a relational and network database concept. Relational database structures, such as those implemented by National CSS and Tymeshare, are useful in handling end-user oriented, "what-if" -type questions. But relational databases require associated machine architectures in order to give users adequate performance.

In summary, the user consensus seemed to be that IMS/VS is of a quality ranging somewhere from "poor" to "ghastly." There is real concern within IBM regarding the poor image of IMS and its lack of flexibility, since the corporation looks to database management and migration as a major tool to fuel future hardware revenue growth.

In contrast, user reports and independent studies or surveys completed by the Datapro Research Corporation indicate that Sperry Univac's

DMS-1100, Burroughs' DMS and Honeywell's IDS offerings, which are compatible with Codasyl (a proposed database standard), are deemed far superior to IBM's IMS/VS.

Potential Remedy

While IMS/VS may be considered the most exposed of IBM's several Achilles' heels, it is also more than likely that IBM will solve its database problems. However, the 360 time-sharing development fiasco suggests that *even* IBM's power, at times, cannot overcome its failures.

A potential remedy could be the incorporation of most of IMS's functions into a new release of the MVS operating system with micro-code (firmware) performance improvements. A number of competing database development projects are under way within the corporation.

Projects with code names such as SEQUEL (Structured English Query Language) and QBE (Query By Example) were first reported in 1974. A recent summary of E. F. Codd's work on relational database structures appeared in IBM's Research Highlights (November 1976). Code names such as XRAM, XQUEL and others are known to exist.

While the final outcome of the ongoing inter-divisional debate over IMS cannot be accurately forecast, it is clear that any new database offering replacing IMS cannot be fully implemented until later in this decade, perhaps in the timeframe of System/80. At that time, a truly efficient relational database product should be available which will resolve most of the user concerns. In the meantime, the major competitors, such as Sperry Univac, Burroughs and Honeywell, continue to enjoy an advantage in the all-important data management area.

Application Generators and Beyond

By 1980, a "new style" of applications development should be available which will help resolve some of the current major problems in usage of computers and computer networks. The growth of the computer industry will certainly be stunted if the present methods of applications development are continued.

The demand for experienced applications analysts and programmers will clearly exceed the supply if the present software "craft" methods are allowed to flourish. Solutions must be found to eliminate the present, significant discrepancies between *what end-users really want and what the analyst or programmer thinks they want.*

Moreover, in today's environment, the presentation of identical applications requirements or specifications to individual implementors results in many different and unique solutions and designs. All too often

the "software wheel" is reinvented — over and over again. Furthermore, programmer productivity declines precipitously in direct proportion to the size of a program (see Figure 2-3 from *The Mythical Man-Month* by Dr. F. P. Brook, Jr.). An applications generator approach could drastically reduce the above problems in the structural design of applications.

Clearly, many of the applications systems have already been defined over and over again, and many of the characteristics of the data they use are known. Proper attention paid to applications generation will provide incredible returns.

Beyond any shadow of a doubt, IBM (as well as several other major manufacturers) are currently developing application generators for the "System/80 era." An example of this is reported in the *IBM Journal of R&D*, Vol. 20, No. 4, July 1976. In fact, one of the key reasons for redirecting the so-called "FS" (Future System) project was reportedly the lack of a solution to the applications development problem.

The new generator techniques will include "transaction descriptions" created by professional programmers and "transaction menus" designed for end-users offering many alternative selections.

As reported in the Wiley-Interscience report ("Data Processing, 1980-85 — Study of the Potential Limitations to Progress"), "the generator would view each application as a hierarchically organized set of building blocks that are to be tied together (binding) in a manner dictated by the choices offered by the generator and selected by the developer (application implementor)."

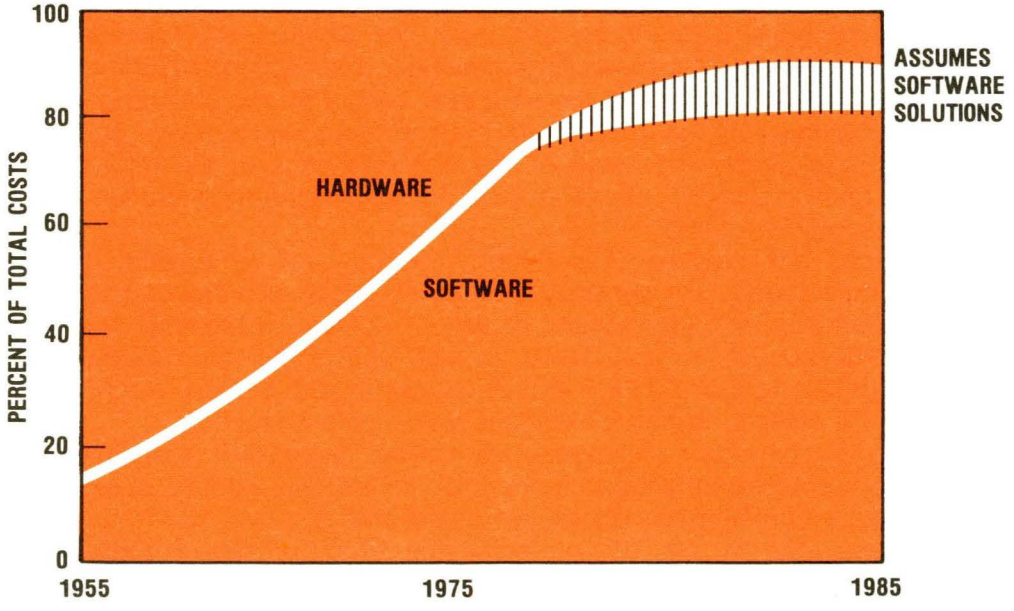
Generators will be provided for different classes of applications (inventory control, payroll, accounts receivable, for example) for specific industries or businesses as well as across industries.

Macro-generative coding techniques are held by many to be the key to quasi-natural language development. Powerful macro-processors allowing language experimentation without the high cost of compiler development are being worked on with very promising results (e.g., Advanced Computer Techniques Corporation's machine-independent macro-processor, HELP II).

From micro-mini to super-maxi, today's hardware innovations by themselves offer no real solution to the data processing effectiveness problems that users have been encountering. For example, the creation of software to really allow distribution of a system has not occurred.

Users beware! If you are having difficulty in the proper use of one system, distribution will probably compound the difficulty. Adequate single-site software facilities are on the way. When they arrive, distributed systems will mean much more than mere hardware distribution running less-than-adequate software at each site. Manufacturers will spend a lot of money in the next years to upgrade software to take advantage of today's hardware.

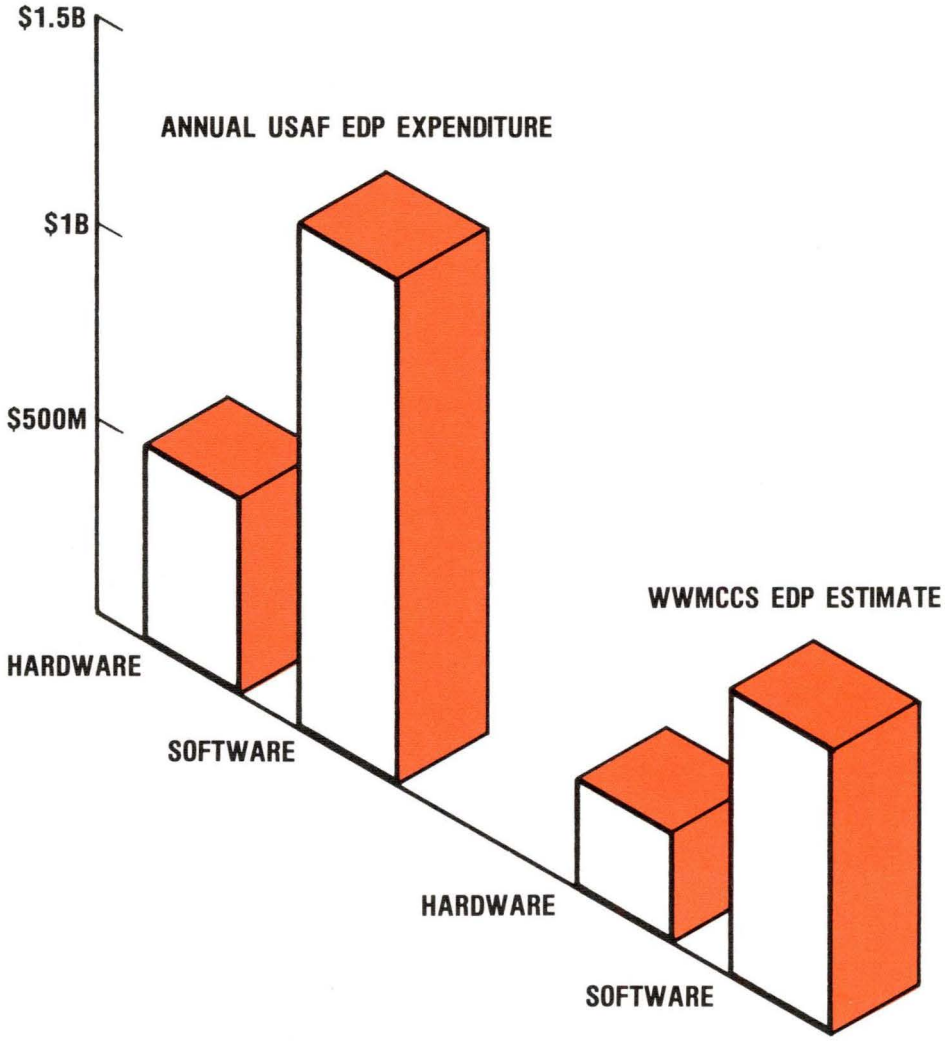
Hardware/Software Cost Trends



SOURCE: Boehm, B.W. 1973 Datamation adjusted by ACT

Figure 2-1

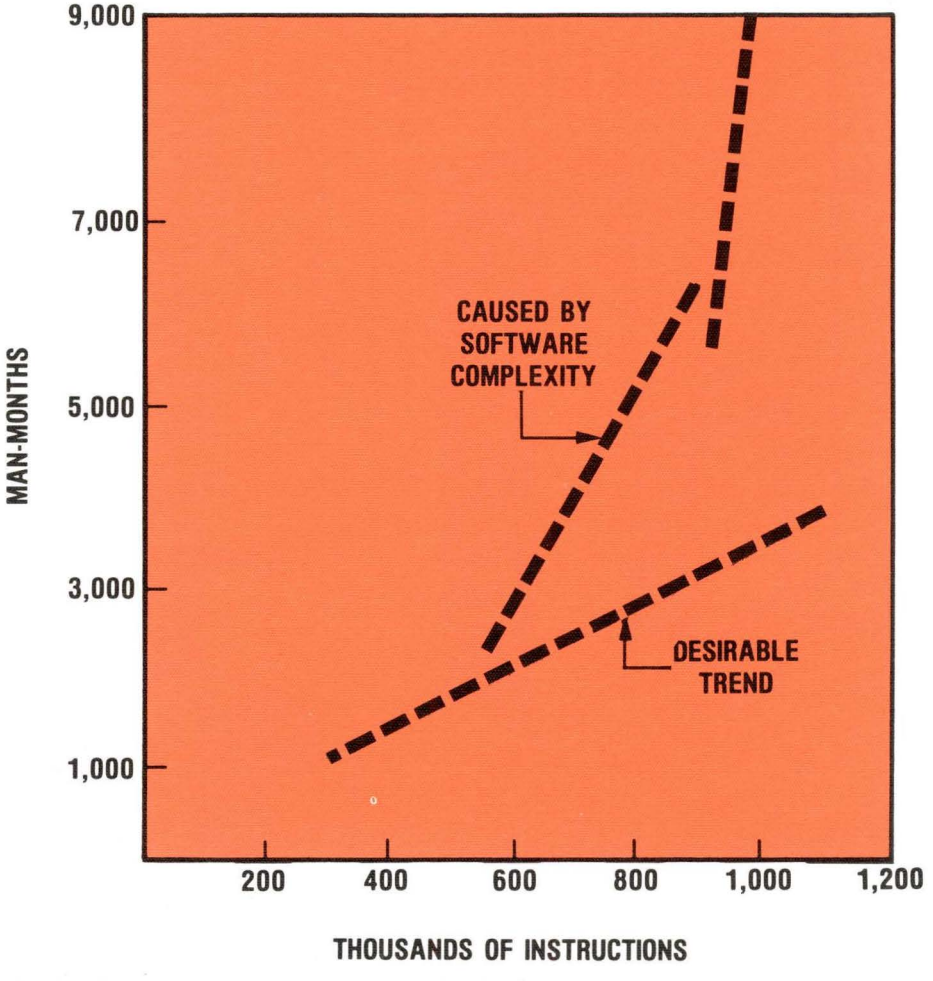
Software is Big Business



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 2-2

Software Production



SOURCE: The Mythical Man-Month — Dr. Fred Brooks, Jr.

Figure 2-3

3. PRODUCTS, SERVICES AND MARKET PLACE STATISTICS

All market indicators clearly point to a powerful trend: continuing, vigorous growth in nearly all segments of the computer industry.

The value of installed computer equipment is expected to nearly triple over the next ten years. The 1976 International Data Corporation (IDC) survey of user spending for hardware over the near term indicated an increase of approximately 17% in 1976 compared with a 12% increment in 1975. (Growth in the software sector has been discussed at some length in Chapter 2.)

At the same time, the nature of the computer industry is changing dramatically; general purpose systems in many sectors of industry are increasingly being viewed as "no purpose," because of the high degree of specialization taking place in product offerings.

Thus, for example, general purpose computer shipments, especially of the IBM 370 series (Models 138, 148, 3031, 3032 and 3033), should peak in 1979 and then begin to taper off in anticipation of new IBM offerings in the 1980 period, specifically System/80.

While this projection alone does not substantiate the no-purpose postulate, numerous changes in the character and focus of DP-related technologies are nonetheless highly suggestive of such a conclusion. The industrially/functionally oriented processor is becoming a reality, however unclear it may be at this point in time. It will, however, not obviate the need for continuing production of general purpose systems for a long time to come, especially in the large class.

Typical Product Cycle

Figure 3-1 shows the typical product cycle for the IBM 360, 370 and

part of the pending System/80 era. The IBM System/360 era began actively in 1966 and ran for about six years. The 370 era was temporarily stalled in 1970-71 because of the recession, and then effectively relaunched in 1972 with the announcement of Virtual Storage (370/VS) systems. This "era" is expected to last until 1979-80.

The peak of System/360 shipments occurred in 1969-70 with approximately 18,000 units shipped, while System/370 shipments should have peaked by the end of this year, with a subsequent rapid decline by 1980. It may be observed that the product cycle for each major system generation has lengthened from six years (for 360) to seven or eight years (for 370) and is projected to expand to ten years for the pending System/80.

Worldwide shipments of U.S.-manufactured general purpose (GP) systems (Figure 3-2) should approach 35,000 units between 1977 and 1978 with an installed (if sold) value of more than $12 billion in 1977 and nearly $15 billion in 1978. It is worth noting that this remarkable set of figures does not even take into account the shipments of mini- and microcomputer-based systems and terminals.

Worldwide terminal shipments of all types (Figure 3-3) surpassed the one million unit mark in 1975, and are expected to exceed 4.5 million units in 1980 and reach roughly eight to ten million units by 1985, provided current trends are maintained. This represents an average of approximately 6.5 terminals per medium-to-large-scale GP system in 1975, and nearly 15 terminals per system in 1980.

It is estimated that 40% to 45% of all the terminals shipped in 1975 were "applications-specialized" in view of their intended specialized usage as point-of-sale (POS), electronic funds transfer, retail, supermarket, and brokerage terminals and so on. The other 55% to 60% were considered "general purpose" terminals of all types.

It appears probable that applications-specialized terminals will achieve quantitative dominance by 1980, with 50% to 55% of all those installed falling somewhere in this category.

Figure 3-3 also shows that approximately 155,000 computers of all types including minis (but excluding microcomputers) were installed by year-end 1975; this volume is projected to nearly double to 300,000 units in 1980. Assuming that the major manufacturers can solve software bottleneck and training issues, one can extrapolate (if current trends continue) to a total of nearly 500,000 such systems installed by 1985.

By 1975, there were an estimated 95,000 customer sites, which represents an average of 1.5 computers per site. There should be at least 160,000 sites (or nearly two computers per installation) by 1980. The number of user sites should approach 240,000 worldwide by 1985.

Figure 3-5 presents a general overview of eleven key product categories, and illustrates the fact that certain competitors are not partici-

pating in some of the high-growth areas; for example, IBM and Sperry Univac were not active in the minicomputer or OEM mini category (although IBM started to become more active in early 1977 with its new Series/1 line, and Sperry Univac recently acquired Varian Data Systems' mini operation). So far, in the important communications multiplexor/concentrator or nodal processor category, Sperry Univac, NCR and International Computers Ltd. (ICL) appear to lack products, although Sperry Univac has introduced one recently; and in the large-scale computer class — IBM 370/158- and 168-type machines — NCR has only recently announced an entry, while the Digital Equipment Corporation Decsystem 1099 or 2050 only reach up to about the IBM 158-3 MP performance level. (DEC's new VAX 11/780 computer line [32-bit word] will likely cause the company to become a full-fledged EDP vendor.)

In applications-specialized or point-of-sale (POS) terminals, Honeywell and DEC are viewed as nonparticipants. Furthermore, very few of the major computer manufacturers offer microcomputer boards or IC components for sale.

Each computer manufacturer is confronted with a decision as to which markets suit its abilities and goals from the standpoint of growth rates, profit margin potential and the "penetrability" of these specific market areas. Figure 3-5 also shows the approximate annual growth rates versus profit margins for each of the major classes of components, terminals and systems offered in the industry today.

Software and applications services represent a very high growth rate area with above average to excellent profit potential. Minicomputers which are sold to the end-user on an OEM basis have average to above average profit potential and high growth rates (in the 30% to 40% compounded annual growth range).

Intelligent terminals and general purpose programmable terminals also offer high growth rates (35% to 40% per year), and average to above average profit potential. The very small business computers, such as IBM's System/34, Sperry Univac's BC/7, NCR's 8100, HIS' Level 6, Burroughs' B80 and DEC or DG products offer below average profit potential in end-user sales (excluding OEM or systems house sales) because of enormous marketing support or "hand-holding" requirements, but the growth rate per year is fairly high (25% to 30% compounded annually).

In contrast, the sale of OEM IC components and microcomputer boards is judged to have poor profit potential because of the many competitors in the marketplace and the "pricing wars" that have occurred and are likely to continue to occur; however, the unit growth rate per year is still very substantial.

At the other end of the spectrum, large computer systems, such

as the IBM 30XX, and other competitor models of that class, offer excellent *profit* potential for modest volumes. The foreseeable growth rate per year is expected to remain fairly high (about 15% to 17% until 1980). This points up the reason for increased competition from Amdahl, Fujitsu, Itel, Hitachi, CDC, etc.

Software Opportunity

Another major growth opportunity is the software-for-a-price business. This can be broken down into three major categories: industry applications packages, general or cross-industry applications software and systems software.

We estimate that the market for all three categories is growing at close to 50% a year (revenue) through 1980. (For example, IBM's revenues for all program products *and* unbundled services probably accounted for roughly 8% to 10% of the total data processing revenue pie, or around $1 billion by year-end 1976.)

With the introduction of such products as IBM's System/34, Honeywell's Level 6, Olivetti's A7 and others, the market for industry applications programs and systems software, as well as supporting services, should now be entering a boom period of at least four years' duration.

Selective Services Growth

The service bureau business, otherwise known as network information services, should also show impressive growth; however, this growth will be selective. In the next five years we should see a role reversal between the two major network service categories — that is, time-sharing and remote job entry — with the latter emerging to account for over one-half the dollar volume for 1980.

This will be attributable to the increasing importance of the remote job entry-oriented systems, electronic funds transfer systems and point-of-sale applications.

The recent introduction and wide acceptance of numerous cost-effective small business systems, minicomputers/microcomputers and intelligent terminals may encourage many corporations to develop their own *in-house* time-sharing or service bureaus with remote distributed small computers or even programmable terminals. If this occurs, as we expect it to, then the traditional service bureau business (as we know it today) may be somewhat impacted.

Counterbalancing this trend will be the emergence of significant

numbers of small users with little or no in-house software capability. They will require outside assistance in obtaining standard, nonstandard or specialized software for their small systems such as IBM's Series/1 and System/34, the Burroughs B80 and the Sperry Univac BC/7.

Honeywell is (among other things) rapidly customizing its Level 6 with Ultra-Text* software to accommodate the newly emerging flood of word processing/management-workstation requirements (product areas which will command our special attention later on in this work). Low-end users will require such customization or tailoring of applications packages and support which, as presently supplied by many of the major vendors, is inadequate.

Differences Fading

Finally, the differences between the computer services vendors and the equipment vendors are starting to fade. The services vendors are enlarging their scope of operations to provide full ranges of products, including hardware, software and specialized support.

Control Data Corporation offers the most obvious example of the broadening of hardware manufacturers' services. CDC and Xerox Data Systems (which was absorbed into Honeywell's computer operations) were among the first to give up the practice of subsidizing software development costs by hiking hardware prices. IBM, too, has demonstrated its acceptance of the premise that software is indeed a proper business for a computer manufacturer to be in; in fact, a growing percentage of its revenues will now come from this single source, and in the 1977-1980 period we should expect most of IBM's supported major operating systems (SVS, MVS and VM) and other software which has not as yet been unbundled to be separately priced (see Chapter 2).

The major software growth areas, with an estimated annual revenue compounded growth rate exceeding 25%, include database management and specialized industry software (which currently accounts for nearly one-half the total dollar volume). Packages in this category include Total, HELP (and its derivatives), IMS, CICS, GIS, RACF (data security), and others.

Fate of Vendors

If the industry as a whole prospers along the lines we project, can all the computer vendors expect to prosper, too (*and* coexist with IBM)? The answer is obviously no.

---

*An ACT Corp. trademark.

There are some who are of the opinion that IBM's worldwide market share (units, not value) is dropping more quickly than might be expected in view of the plethora of minis finding their way into the marketplace and the new vogue of linking these into distributed networks. Were this the case, it might accelerate the demise of some of the other major manufacturers whose pricing policies presume a high IBM umbrella. It should be interesting to see how the Justice Department handles this "Catch-22" situation.

Some of the major manufacturers, such as Burroughs and Sperry Univac, have been adding to their installed base at a rate of over 15% per year. Sperry Univac reports that over one-half of the 1,000 orders received for its 90/30 system by year-end 1975 came from *new* customers, and it expects to increase its state and local government base at an annual rate of 13% through 1980. (In fact, Sperry Univac sold its 2,000th 90/30-25 in fiscal 1977.)

The state and local government market is one that is still relatively unexploited; there are others. A well-publicized study from International Data Corporation (IDC) lists 18 key industries, each of which has a data processing spending potential of over $200 million per year. The largest of these markets, in the order of their potential growth rates, are: hospitals, commercial and savings banks, automotive and other dealers.

"Electronic Offices"

No discussion of today's marketplace trends would be complete without at least a few words about the "word processing" or "electronic office" market. IBM vice-chairman, Gil Jones, stated at a security analysts' meeting that by 1985 the company's word processing operations, i.e., its Office Products Division (OPD) and General Systems Division (GSD), would be equal in size to its data processing operations (DPD) today. In other words, by 1985 this business alone should yield $10 billion or so in annual revenue.

IBM's new Word Processor/32 product is a hybrid providing both data processing as well as word processing capabilities. The characteristics of the new product line indicate that inter-company electronic mail applications are around the corner.

In the U.S. alone, the GSD has more than 2,000 salesmen who will be working with the OPD in focusing on the "office system" or word processing market. OPD will still have primary responsibility; however, included in the Word Processor/32 product is the new ink-jet printer (Model 46/40), an unattended magnetic card-to-printer or hard-copy machine. The ink-jet printer produces typewriter-quality pages at a speed of 77 to 92 characters per second, roughly two-and-one-half times that of equivalent competitive offerings.

Xerox, Burroughs, DEC and Honeywell are also in the so-called word processing market, but the latter (in collaboration with BASE, a subsidiary of Advanced Computer Techniques Corporation) is producing a product which goes much further than the IBM word processing/data processing offering. Cluster systems including electronic mail, electronic file cabinet, indexing and information storage and retrieval are in their plans for 1977-78.

The true management workstation environment (of which word processing functionality is but a subset) is indeed just around the corner. Many companies, both in and out of the traditional computer industry, are serious contenders for a marketplace share in this ever-growing arena.

Finally, terminals (which can be part of office systems) represent another high growth area for the 1980s. Users should require about 4.5 million terminal units by 1980. At present, there are over 1.3 million units installed. This amounts to an annual unit growth rate of approximately 25%.

General purpose and applications-specialized, programmable or high-IQ (potentially intelligent) terminals will enjoy the highest annual growth rate, in excess of 40%. In terms of the variety and quantity of terminals appearing on the scene, the ensuing years will be tumultuous indeed, and the question of terminal support will become all the more complex as a result. We will discuss this market in a later chapter when we address emerging trends in communications.

An industry growth-inhibiting factor of the data processing marketplace identified by Advanced Computer Techniques Corporation (ACT) in the course of its industry reviews suggests that a major, fundamental roadblock to the use and growth of data processing is the lack or limited availability of data to the end-user.

Over 80% of all data processing today is believed to be sequential and batch-oriented. Computer users often maintain many tape files in which essentially inactive and/or redundant data is allowed to remain to no constructive purpose.

Such elementary drawbacks in prevailing systems design concepts must be kept in mind as we seek to focus our energies properly in meeting the needs of the 1980s. The systems of the next decade should not and need not be compromised through the perpetuation of current design insufficiencies, and the question of improved access methods is a prime candidate for our attention.

Certainly, more processing power will be required and significantly greater CPU "horsepower" will be provided (although a single processor "engine" by 1980 will probably not achieve more than two to three times the IBM 3033 level of performance because of the inherent heat and speed-of-light limitations of current high-speed LSI technology). The I/O channels or their equivalent (I/O processors) will also have increased

data rates. Main storage will undoubtedly be substantially (ten times) larger and faster with more functional designs (for example, hierarchical storage structures).

By 1980, "small" machines will be able to offer memory sizes comparable to today's large-scale computers, allowing the same job and common operating system to be used across all compatible models from the low-end to the very high-end machines. Furthermore, the old arguments regarding economy of scale ("Grosch's Law") which currently underlie the tendency to install some applications (primarily scientific types) on centralized large computers, will continue to fade away as the cost of the CPU, and especially of memory, erodes.

Key Functions to Optimize

The real answer to the question of what to improve is that computer manufacturers should dedicate at least part of the greater processing engine power to providing easier, faster access to data, rather than using it merely as a means to house ever-larger collections of data. Another key function to optimize in systems to be delivered in the early 1980s concerns the *format* of the data whose access we seek to make more efficient.

Data is the most essential element in this kind of system orientation, the "new center of gravity." The so-called arithmetic or central processor units (CPUs) will no longer be "central" — they will be "slaves" (away from the center or on the periphery of the system).

The real growth in data processing will not be in the amount of data, although, to be sure, this will grow substantially, but rather in the *usage* of that data. Control over data should be another prime candidate for optimization. Today, the inability to share, or the inefficiencies involved in sharing, the database of a given enterprise is the single greatest inhibitor to the usage of that database. Users can't easily get at the essential data which *already exists*. As a result, they invent their own private copies or subsets of the fundamental data.

Much of the complexity of systems stems from the unavailablity of appropriately formatted data when it is wanted. To resolve this major inhibitor to application (and thus to industry) growth, therefore, new, powerful, LSI "process engines" must be explicitly optimized and adapted via microcode to accommodate a need for significantly increased data accessibility.

In our view, this implies a radical adjustment in the way we think about data organization today (but not so big an adjustment to how we thought about it in the 1950s). Rather than designing data files from the perspective of a given application, we may well learn to think "from the

bottom up"; that is, to construct a global database in which all constituent data elements are logically ordered in terms of their names and interrelationships prior to any definition (and thus circumscription) of their use.

Specific applications of great diversity can grow readily from such a source, because the limitations that formerly conditioned the organization of the data — the particular exigencies of a unique applications goal — shall not have been a factor in shaping the database design concept. The probable differences in perception of the data among variously motivated applications programmers and end-users will not impede their common effective use of the database that has been constructed in this fashion. Data relationships will be clearer and applications emerge more easily than they do within the limits imposed by our current orientation. (IBM, in its new technical releases, refers to this new "data" orientation as a "system resource repository and controller," or as "the outboard data manager.")

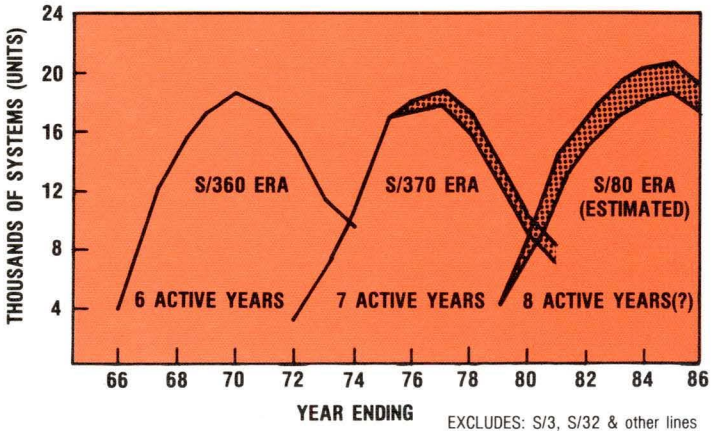
The actual implementation of this concept (also at times known, at IBM, as "relational data structure") will require a lot of engine power. However, overlapping functions and interdependencies will be sharply curtailed or even eliminated by this approach to the future system, and the true *value* of data will be commensurately enhanced as timing and accessibility receive their enlarged share of emphasis in the new design concept.

The growth and vitality of the computer industry hinges on the direct commitment we make to this transition. Also, database machines composed of distributed minis, each controlling a sector of the corporate database, should forever lay to rest the need for database conversion through plug-compatible software to central site CPUs.

The changes of which we speak here must be incremental, and the redesign of current applications systems and data formats will be gradual. Old batch mode applications will continue to run without change; indeed, some may never change, as they run on COBOL, FORTRAN, etc. processors linked to new systems technology — in homes for the aged. Those applications that will change will find their way gradually into the new systems, without impacting current production systems.

Systems do not have to be changed in zero time in order to take advantage of new processing technologies. But eventually the data, the processing and the end-user accessibility that together comprise a "whole" system will be controlled by an intelligent "system resource repository and control box." We wait breathlessly!

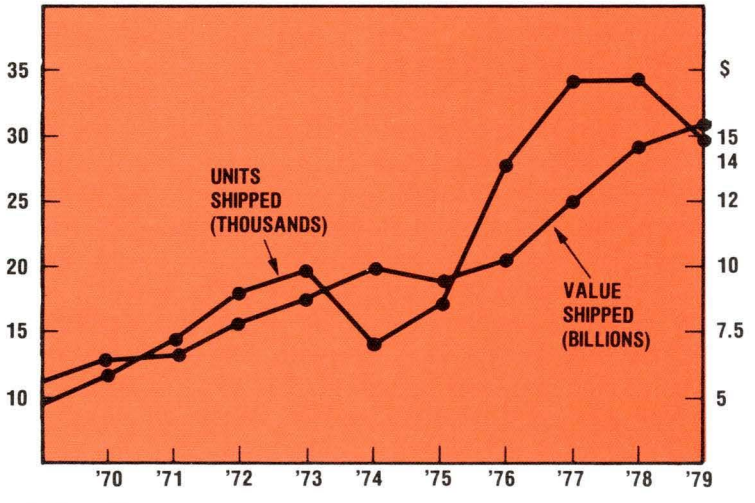
Total Shipments



SOURCE: ACT/Technology-Analysis Group (TAG)

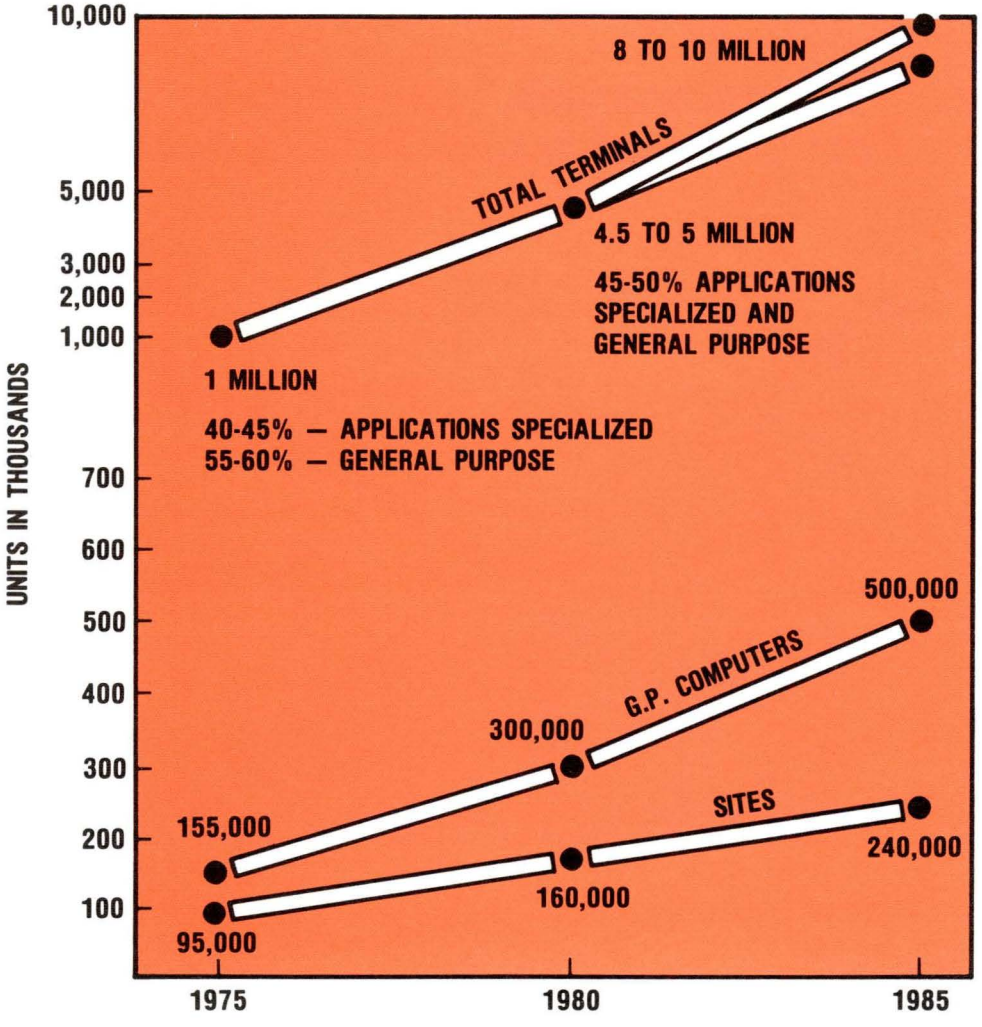
Figure 3-1

U.S. Mfg. — Worldwide Shipments of G.P. Systems



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 3-2

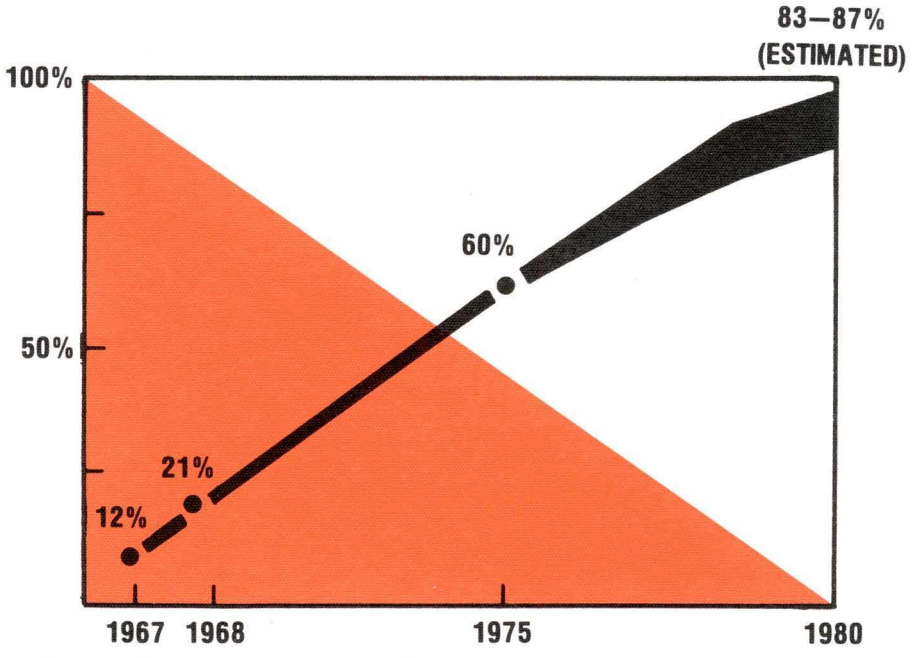


SOURCE: A.C.T. estimates

SOURCE: ACT/Technology-Analysis Group (TAG)

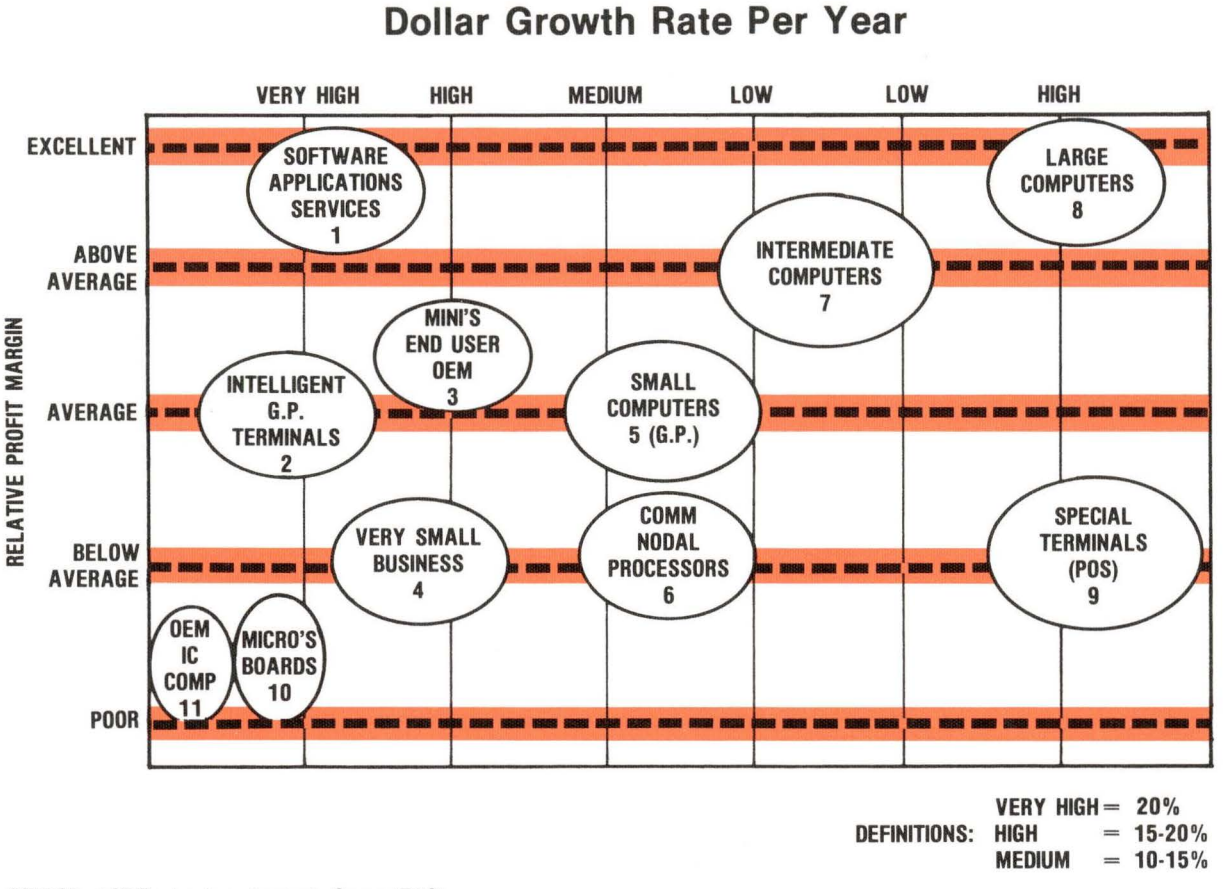
Figure 3-3

Communications Technology Trends
Data Communications-Oriented Systems



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 3-4



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 3-5

4. MANUFACTURERS' SPENDING EMPHASIS AND TRENDS

The major data processing manufacturers face an expanding range of possibilities in planning the investments and disbursements that will establish them in the most profitable and promising markets of the 1980s. The difficulties that they must overcome in planning their spending strategies do not arise from lack of information concerning individual markets, although the quality of that information may be questionable in certain areas.

Rather, their problems stem from the need to decide *which* of those markets are the right ones for them to be in. Every market and product decision entails its own risks, for which a company, by reason of its technical, financial and/or other limitations, may prove to be inadequately prepared.

Apart from an IBM or an AT&T, few manufacturers can afford the luxury of both expending their resources on multiple products in nearly all market areas, and of generating and maintaining "second front" plans/products with which to meet all contingencies as well.

Expenditures and Potential Opportunities

General purpose computer systems as we know them today, in the small-to-large-scale areas, are expected to grow at an annual rate of 17% nationally and nearly 15% worldwide for the next few years. In contrast (as stated in Chapter 3), general purpose programmable terminals are projected to enjoy a 30%-35% compounded growth rate. Also, specialized point-of-sales (POS) or electronic funds transfer systems (EFTS), among others, should climb at a 35%-45% dollar growth rate.

Small business computers such as the Sperry Univac BC/7, NCR 8100, Burroughs B80, IBM System/34, Honeywell Level 6 and others

produced by such worthy contenders as DEC, DG, Interdata, Hewlett-Packard and Olivetti, should grow at a rate in excess of 30%.

The relative distribution of these growth rates suggests at least the possibility that some manufacturers may be in the wrong market. For example, they could be concentrating most of their resources on the general purpose mainframe area, which has a relatively lower growth rate, or dissipating their limited resources in a number of areas with no clear idea of where to focus them.

While a given market, such as that for general purpose and applications-specialized intelligent terminals, may promise a relatively high growth rate, it is also likely to be characterized by intense competition and rapidly falling prices. Thus, there are present all the elements of a Waterloo for a firm which lacks sufficient financial staying power or research and development resources to compete in this arena.

As a further example, the small business systems and minicomputer end-user market has a relatively high growth rate; however, it also involves a great deal of customer hand-holding, marketing expense and software investment which few firms are capable of handling profitably. Ostrich-like, first-time users may ultimately be the victims of naive entrants into this marketplace.

Research and development budgets usually (but not always) reflect individual corporate decisions as to which market and specific budget(s) or business(es) to be in. At the same time, R&D budgets are frequently used for purposes not necessarily in the interest of the parent company.

Lack of emphasis on or adequate spending levels for promising equipment or applications, overly narrow goals or excessive experimentation are some examples of the inherent risks. Also, triage seems a common method of investment planning. Thus, many projects with initial financing are left to wither as committed funds are exhausted.

Finally, the guillotining of worthy projects prior to their maturation as a result of management jitters is a more prevalent mode of decision-making than most would care to admit. Univac's recent cancellation of its Unified Product Line (UPL) project is a striking case in point.

Without being privy to the internal budgets of the various manufacturers, we may reach certain conclusions from the recent flow of product introductions and other information in the public domain. IBM's apparent number one priority is I/O devices, such as disc files, laser printers, mass storage subsystems and so forth, while number two is probably software, including operating systems enhancements, new operating system development and applications software.

The third-ranked priority would appear to be mainframes, covering all categories from the System/32 and the 5100 all the way up to the super-large-scale 3033 and its about-to-be-announced multiprocessor versions. Number four is likely to be components development, including

logic and memory circuits and packaging. Number five is the new communications satellite venture, as well as terminals, multiplexors and concentrators. Number six, the "all other" category (also a major growth area), includes office products, word processing, copiers, etc. Finally, in our estimation, priority number seven at IBM is pure research; it will come as no surprise that few, if any, other vendors can afford to invest the estimated $100 million (or 10% of a total $1 billion R&D budget) on *pure* research that IBM is believed to have spent in 1976.

While communications is of relatively low rank at this writing, it probably has a higher growth rate in R&D expenditures at IBM than the other categories.

Similarity of Burroughs, Sperry Univac

In contrast, Burroughs apparently is spending most of its limited R&D resources on mainframes of all types, e.g., the B800 series, followed by communications terminals and controllers. Its third-ranked priority is evidently software, including new MCP operating system versions and business applications packages, with I/O devices probably ranking fourth and "all others" (including word processing) fifth.

Components represent a relatively small current R&D activity at Burroughs, but nevertheless an important one — about 30%-40% of their logic and IC storage needs are internally met.

Sperry Univac's key priority (like Burroughs') appears to be mainframes, followed by software, where it reportedly is spending in excess of 50% of its R&D budget (on operating systems extensions and system software enhancements, as well as other essential packages).

Priority number three is probably I/O devices (at its Information Storage Systems Division); number four is believed to be communications and terminals, including network processors and remote concentrators.

Finally, number five is likely the components area. Sperry Univac, like many other vendors (excepting IBM), depends to a great extent on outside suppliers for most of its IC components. Since developments in IC component technology are occurring at a rapidly accelerating pace, the absence of significant internal investment in IC manufacturing may prove to be an advantage — or at least not a disadvantage.

Emphasis at Honeywell, DEC and NCR

Lately, Honeywell seems to be giving greater attention to applications and operating systems software, while its second level of priority appears to be occupied by mainframes of all types, including minicomputers

and word processors. Probably next in order is I/O devices, although Honeywell gets many of its magnetic and other media-type devices through its joint peripheral venture with CDC and ICL. Communications and terminal emphasis is followed by component development, with modest R&D funds spent on pure research.

Digital Equipment Corporation is clearly spending most of its R&D budget on mainframes to support its dominant position in the minicomputer market (and its new, larger scale VAX line), followed by a more recent emphasis on software. I/O devices represent the third most important area, followed by communications and terminal products. More recently, DEC has been improving its vertical integration by developing in-house components capability.

Unlike the above-cited domestic manufacturers, NCR has continued to place its emphasis on POS terminals for its traditional banking and retail marketplaces, while reinvigorating its processor technology. Its introduction of a complete new line (the 8000 series) would suggest that the company intends at least to maintain its market share in the small-to-medium-scale mainframe business. Of equal importance to NCR lately is its software situation, which is commanding a greater share of its management's attention and R&D funds.

The above six characterizations cover more than 90% of the computer systems manufacturers' R&D expenditures being made at this time. The omission of such noteworthy firms as Amdahl, Fujitsu/Hitachi, Data General, Olivetti, Siemens, ICL, etc. from the above discussion is intentional, but in no way designed to reflect upon their importance.

For one thing, most of the larger foreign companies are subsidized by governmental monies and their budget priorities are therefore difficult to identify. As for domestic companies, further data would not significantly alter the overall trends already described.

If one were to add up all the 1976 research and development expenditures of all the above-listed vendors (and many others), they would not equal IBM's R&D expenditures for the same year. While this is an impressive (if not entirely unexpected) statistic, it is perhaps less significant than the implication that IBM is capable of continuing new product/new version introductions, which in turn compels the others to react, thereby dissipating their already meager resources.

IBM's vast market research and product planning organization is better acquainted with the general user requirements for the 1980s than anyone else. Since its development arm participates very actively in most user meetings which occur throughout the world semiannually, and its market research program is so extensive, it is better equipped to ride the waves of change.

In fact, in view of its dominating position, IBM may be creating the very waves it rides. While few other vendors can afford a sizable invest-

ment in marketplace analyses, the question is, can they afford not to? It would appear that selectivity and depth are preferable to attempting to cover the whole industry.

When G.E. announced its intention to compete across the board with IBM in the late '60s (just before it sold its business to HIS), this tended to reconfirm, in the minds of some industry watchers, the validity of Russell's remark, to wit: a hero is a person who hasn't examined all of the facts.

For example, some competitors today are unnecessarily wasting precious resources on the enhancement and maintenance of old operating systems and other software. Many of these are, as McLuhan observes, looking forward through a rear view mirror.

Other data processing suppliers may be too narrowly focused — spending most of their funds on developing special-purpose terminals or other more exotic, nonstandard and low-volume products. Still others have spread themselves too thin, dissipating their resources on every type of exotic device, some of which may, in fact, be important, but others of which are of less moment in the marketplace. As a result, they lack sufficient funds and the requisite "critical mass" to accomplish any single major goal.

Distribution of the Hardware Sales Dollar

Obviously, IBM enjoys cost advantages at almost all stages of its activities. This is most noticeable in manufacturing, where economy of scale is critical.

IBM's advantage is even greater where its software sales dollar is concerned. Its internal software development costs are spread across so wide a customer base that profitability of any single product is almost guaranteed. This is especially the case in view of the cheapness of manufacture of completed software products, whose prices reflect value received rather than production cost.

Faced with IBM's greater volumes and market share, other computer vendors really have only one course to follow: to divest themselves of all nonessential activities, and redirect and focus their limited resources to support the key aspects of their businesses.

Figure 4-1 is a summary of the breakdown of the IBM 370/135 mainframe costs based on the IBM Graybook which was an exhibit in the Telex trial. You may note that manufacturing represented about 18.5% of the cost (as a percent of revenue). We estimate that the new model 138 with the lower cost MOS (instead of bipolar) storage allowed IBM to reduce this cost to about 12.5% of revenue.

The bottom line indicates that IBM's total costs for this product

were 72.4% of revenue, declining to roughly 63.5% for the new model 138 with an estimated profit before tax of 36.7%. This is consistent with speculation that IBM may at times have been operating at 40% profit (before taxes). In 1976, they reported nearly 30%.

Manufacturers are beginning to de-emphasize hardware engineering and production (see Chapter 2). This has been caused not only by a shift of emphasis from hardware to software, but by their realization that some low-volume products can best be bought outside from specialized concerns. The NCR, CDC and ICL peripheral activity is a good example of this sort of shift in emphasis. We would expect other, perhaps communications terminal-oriented, ventures of this ilk to flourish also, in the 1977-80 period.

Better software and services are far likelier to sell computers these days than higher processor speeds. Firms which do not have IBM's current billion dollar research and development budget, take note!

Configuration Deficiency

A glaringly obvious deficiency in the quality of support frequently occurs even before the vendor/customer product relationship is firmly in place: the failure of salesmen or analysts in the field to properly configure the systems they propose, i.e., to fit the applications or performance levels to the requirements of the customer.

NCR's Computerized Order Preparation and Network System (COINS), developed by Stephan Rothschild of the General and Applications Software Development Department and recently touted in the news media, is one of the first on-line systems available to a sales force (NCR's own) to support better configuring.

Surveys have indicated that greater user support is badly needed at all levels. This need, however, is most obvious at the low end of the business computer area. There, a desire exists to leave applications software to local software houses and supply only essential systems software or "tools."

This was apparently the strategy of DEC and others although, according to some analysts, DEC is not yet really a *full* participant in the "business computing" market, no matter how often or how wistfully it talks about it. The impact of the new VAX line's introduction may justify some reevaluation of DEC's position in the future, however.

IBM Forecasts

The table in Figure 4-2 shows IBM's forecast of product life, revenue,

profits and margins for many key items; i.e., 370 mainframes and popular I/O devices. (Some of this information comes from the IBM Graybooks which were part of the Telex trial exhibits.)

Column one shows the variation in *forecast life* for different products, which is considerable (actually there are many product lives; i.e., forecast rental life, pricing life, physical life, economic life and committee rental life). Generally, products with an earlier expected *total* obsolescence will also exhibit a lower forecast life; e.g., the original 3330-1 disc was first shipped (approximately) in July 1971 and actually replaced with the 3350 disc subsystem, shipped in approximately March 1976, or nearly 56 months later (compared with the forecast life of 58 months).

Another example is the 370/145, which was initially shipped in July 1971 and subsequently obsoleted by the Model 148 (which was scheduled to be shipped by about January 1977 — approximately 67 months later or exactly on target). In fact, the 145 rental life is somewhat longer, since 145 customers will gradually upgrade to the 148 (depending on availability schedules), and many will stay with the "old" 145s due to the purchase option credits and recently discounted purchase price.

In addition, Models 158 and 168 have a very low 46-month forecast life because most (estimated at more than 80%) of the mainframes in this class are purchased outright. Assuming IBM's forecast life is nearly accurate for these models, one could easily have forecasted the announcement of a successor to the 158-3 by the third quarter of 1977 (Models 3031 and 3032).

The 168-3 upgrade model, the 3033, was announced in the first quarter of 1977. A multiprocessor version of the 3033 is also anticipated by early 1978.

Column 4 shows IBM's expected pre-tax margins; e.g., 33.8% for the 3330 disc and 40.8% for its 3830 controller, while the 3420 tape drive pre-tax margin is only 26.9% (perhaps reflecting the plug-compatible impact on profit levels). The bottom line of 33.1% is the weighted total margin for all products in this table. In our judgment, these figures are pretty much on target.

A note of caution is perhaps in order. The forecast life should not be taken too literally, as other factors such as competitor developments (Amdahl/Fujitsu, etc.) and planned/unplanned "mid-life kickers" (enhancements) can shorten or extend the forecast rental life of a product.

Recent examples include the attached processor 158/168, 3705 version II, and the 3330-11, all of which extended the product rental life.

System Expectations

The next table (Figure 4-3) indicates IBM's expected revenues, profits and margins for the 370/145 *system*. Note that both *pricing life* data (based upon a 48-month nominal average life expectancy) and *forecast life* data, which is based upon IBM actual expected life forecasts, are shown.

For the total system, note that forecast life revenues of $4.345 billion exceed the pricing life revenues of $3.446 billion by 26%, which implies about a 60-month average forecast life for the *total* system.

Two major 370 planning points should thus be emphasized: (a) the pre-tax margin on pricing life (which is somewhat proportional to constant markup over "manufacturing costs") varied from 22.1% on card equipment to 38.5% on disc equipment, with a system average of 31.3%; and (b) the incremental margins (or the difference between pricing life and forecast life) vary from 43.3% to 58.8%, with an average of 54.1%!

The high profit associated with such long-term risk leasing (i.e., over periods in excess of 48 months) made this activity extremely attractive (understandably) to IBM. Clearly, it would have accounted for a significantly smaller fraction of its planned business if it had felt itself vulnerable either to competition, or even to the impact of one or more of its own product releases!

The probable reason that the incremental margins were not even higher is that the 370/145, believed to be on 60-month depreciation, continued to absorb a substantial (albeit fixed) amount of corporate overhead.

The 138 and 148 announcements and, more recently, the 3031, 3032 and 3033, tend to emphasize long-term leases or purchase; the pricing relationships seem to discourage rental. Apparently rental business is too risky for IBM at this stage, *or* it anticipates that the rate of technological change will accelerate over the next two to three years and it thus wishes to reduce its rental base exposures.

The new large-scale 3033 order rate is believed to have quickly built up to an incredible 2,000 units in less than two months and deliveries are extending to about the third quarter of 1979 (2,000 Model 3033s X > 4 MIPS = 8 BIPS; that is, billions of instructions per second. The estimated 10,000 orders for the 3031 and 3032 bring this to approximately 22 BIPS, and nearly $25 billion in additional revenue by 1980.)

The infusion of this enormous processing power into the world marketplace before the end of this decade may in itself be an overriding factor in molding our industry.

DISTRIBUTION OF IBM MAJOR COSTS AS A % OF REVENUE

IBM 370/135 MAINFRAME COSTS		ESTIMATED IBM 370/138 COSTS
18.5%	Manufacturing	12.5%*
7.2	Customer Engineering Support	2.2**
1.1	Engineering	2.5
0.5	Reconditioning	0.8
0.3	Scrap & Rework	0.5
0.7	Other	0.7
38.7	Corporate Apportionment	38.7
5.4	Profit Contingency	5.4
72.4% =	Total Cost	= 63.3%
27.6% =	Gross Profit	= 36.7%

*Low cost due to bipolar replacement with 2K MOSFET chips.
**CE cost is essentially covered by unbundling charges.

SOURCE: IBM 370/135 Graybooks (Telex Trial Exhibit)

Figure 4-1

IBM 370 PRICING & PROFIT MARGINS

	MODELS	(1) FORECAST RENTAL LIFE (MONTHS)	(2) REVENUE ($ MILLIONS)	(3) PROFIT ($ MILLIONS)	(4) PRE-TAX MARGIN (%)
135	Mainframe*	70	2,034	723	35.5
145	Mainframe	67	2,293	806	35.1
158	Mainframe	46	1,663	578	34.8
168	Mainframe	46	1,255	310	24.7
3211	Printer	67	594	168	28.3
3811	Controller	67	301	108	35.7
3360	Memory	56	760	302	39.7
3330	Disc	58	1,639	554	33.8
3830	Controller	57	959	391	40.8
3420	Mag. Tape Drive	58	1,360	365	26.9
3803	Controller	58	464	108	23.3
	TOTAL		$13,322	$4,413	33.1%

*Mainframe includes — CPU, memory, I/O channels and required features (excluding I/O devices, program products, etc.)

SOURCE: IBM Graybooks (Telex Trial Exhibit)

Figure 4-2

PRICING — 370/145

	REVENUE ($ MILLIONS)			PROFIT ($ MILLIONS)			MARGIN (%)			INCREMENTAL MARGIN (%)
	Forecast Life	Pricing Life	Variance	Forecast Life	Pricing Life	Variance	Forecast Life	Pricing Life	Variance	
CPU, Memory, Features	2,293	1,817	476	806	545	261	35.1	30.0	5.1	54.8
Tape	586	464	122	183	124	59	31.2	26.7	4.5	48.4
DASD (Disc)	776	615	161	329	237	92	42.4	38.5	3.9	57.1
Printer	385	305	80	162	115	47	42.0	37.7	4.3	58.8
Card	143	113	30	38	25	13	26.4	22.1	4.3	43.3
Communications	167	132	35	50	33	17	30.0	25.0	5.0	48.6
TOTAL	4,345	3,446	904	1,568	1,079	489	36.1	31.3	4.8	54.1

SOURCE: IBM Graybooks

Figure 4-3

5. DATA COMMUNICATIONS AND NETWORK TRENDS

IBM and AT&T seem to be pursuing longer term strategies which suggest major competitive confrontations by the end of this decade. Both want to supply a total, end-to-end service based on devices capable of performing multiple functions (as opposed to today's dedicated, special-function devices).

Initially, the key challenge is to define the relative meaning of "data processing" in such a way as to arrive at some workable consensus on the meaning and roles of "message switching," "transmission," "circuit switching" and other, interdependent terms and concepts. This issue is currently being hotly debated by the Federal Communications Commission, the legislators and the various vendors for reasons far more compelling to the various debators than mere semantic purity.

Some developments that can be viewed as early warning signs of the impending, "all-stops-out clash of the giants" including the following:

- IBM, Aetna and Comsat have undertaken a troika venture — Satellite Business Systems (SBS) — to create an all-digital domestic satellite system to handle data, voice, video and other types of communication services.
- AT&T is offering an electronic funds transfer system (EFTS) in the State of Washington, with potential for nationwide expansion.
- IBM has publicly voiced its opposition to AT&T's data processing networks.
- IBM is marketing a PABX system in Europe and plans to introduce an advanced PABX (code-named "Rosebud") domestically soon.
- AT&T has announced that it will soon introduce its competitive alternative to IBM's Systems Network Architecture (SNA) and

Synchronous Data Link Control (SDLC) line protocol. Official product and service announcement of Bell's Data Network (BDN) is expected in the second or third quarter of 1978.

- AT&T has sponsored legislation, entitled the Consumer Communication Reform Act, the essential aim of which seems to be to eliminate the competition in the data communications marketplace.
- AT&T is gradually expanding its digital data service (DDS) to 26 major cities in 1977 or shortly thereafter (Figure 5-1), with the concomitant potential for future technological development and implementation of fiber-optic cables for local hookup (offering very wide-band transmission and high reliability).

Meanwhile, other companies are making their own moves: RCA has successfully launched its second satellite; Western Union is actively involved, as are the value-added network (VAN) companies such as Packet Communications, Inc., Telenet, Graphnet and the special carriers (e.g., MCI).

All this activity is giving rise to a new wave of technological innovation within the communications industry that will reach the user with ever-increasing frequency during the 1978-82 period.

Why the Interest?

Why all the apparently sudden interest in data communications? Quite simply, because the market's absolute size and projected growth rates are enormous and, consequently, enormously attractive to industry heavyweights.

Some statistics will demonstrate its potential:

- Department of Commerce statistics show that telephone and teletype traffic *alone* will constitute a $95 billion industry by 1985.
- By 1985, if current trends continue, it will cost less to transmit a facsimile letter than it will to send a conventional letter through the mail (AT&T claims this will be possible sometime in the 1978-79 timeframe).
- By 1980-81, about 80%-87% of all the computers in the world should have at least one or more remote terminals on-line (Figure 5-2).
- The demand for terminals and other communications equipment is expected to grow by approximately three to four times the rate of the general purpose mainframe market.
- By 1980, more than 30% of the total users' hardware budget should go for communications-related products and services,

with this demand expected to grow to over 50% by 1985.

Of course, the ultimate success or failure of the competitive offerings mentioned above, and the validity of the cited growth statistics, will depend substantially on the outcome of current legal and regulatory issues. Their resolution will dramatically impact the growth of the network and DP markets.

IBM's acknowledgment of the stakes at issue is readily perceived in the scope of its SBS venture and the intensity of its government lobbying activities.

Colossi at War

In nearly all respects (excepting the literal deployment of firepower), the emerging struggle between the AT&T and IBM colossi recalls the spectacle of two great nations poised for war. Let us consider their "weapons resources" while noting the rather striking disparities between them as they probe for advantage.

At year-end 1976, AT&T had about five times the number of stockholders and assets of IBM (or $86.7 billion vs. $15.5 billion, respectively); more than three times the "army" size (927,000 vs. 292,000 employees); nearly twice the revenues (about $16.3 billion for IBM vs. approximately $33.2 billion for AT&T); and over 1.5 times the net income in 1976 (nearly $2.4 billion for IBM vs. over $3.8 billion for AT&T).

On balance, however, IBM is far more profitable than AT&T when the percentage return on total investment is considered as part of the equation. Also, it is generally held that IBM is the better managed of the two contestants, and richer in marketing capability than AT&T.

If you think the outcome of this "war" is a foregone conclusion, the remainder of this chapter provides additional facts which suggest that neither party to the conflict can win — or lose — easily.

AT&T Marketing

The AT&T offering (the SNA and SDLC analogues or BDN) mentioned earlier, likely to be announced by 1978, is designed to provide users with an alternative to IBM's communications modus operandi. Its marketing thrust will be to offer users more flexibility than they have come to expect from IBM, which historically has been wont to *modify* the "de facto standards" it may itself have established, with a view to promoting "exclusivity" (i.e., maintaining technological pressure, quite naturally, to encourage the exclusive use of IBM auxiliary equipment).

The AT&T approach would be universal, in that compatibility with a large number of different products would be provided, assuring users that they would not be locked in by the effects of IBM's anti-establishment posture.

We sometimes forget that AT&T is a public utility governed from within *and* without. IBM, on the other hand, as with any other "free" enterprise, is subject to less restriction. In addition, since AT&T would be providing a *communications service*, various hardware which is currently acquired in toto by the user would be displaced by AT&T central office *shared* hardware, thus at least raising the possibility of reducing usage costs — all other things being equal.

What I really believe is that no matter who "owns the store," higher user costs are inevitable.

Some of the projected characteristics of AT&T's potential service are listed below (based on an AT&T presentation to IBM's users in the spring of 1976):

1. AT&T's offering is not quite a "value-added network" (VAN), but would set out to provide an alternative to IBM's proposed Systems Network Architecture.
2. It would transfer to the telephone company's central office most of the *communications-related functions* that are now performed by central processor front-ends, multiplexors, concentrators, and remote controllers and intelligent terminals (these last currently the subject of a price war).
3. It would be available "within much less than five years."
4. Among other functions, it would provide network control, speed conversion, error control, terminal polling, message routing/rerouting and formatting. In IBM terminology, the system would exclude access method (VTAM) but include network control program (NCP).
5. It would stop short of "processing the information content of bits" (i.e., no data *processing* service). However, as a benchmark, its functionality would exceed that of the famous ARPA network (Figure 5-3), which can only move constructed "packets" of information.
6. A high-level language would allow users to specify their communications processing requirements in standard form.
7. Besides central office equipment, some on-site controllers might be needed.
8. Office automation is one possible example of AT&T's industry marketing thrust; this is because significant changes in the way we conduct office activities are expected to occur (CRTs versus typewriters, printers versus copiers, electronic storage versus file cabinets, all tied together into communications network

systems), setting preconditions for communications network demands (Figure 5-4).

9. AT&T thinks its system will not violate the 1956 consent decree or 1970 computer/communications decision, but this position is obviously self-serving.

It seems that the new AT&T service could affect many different companies in both the data *processing* and data *communications* fields, but it is not yet clear exactly how. For example, it might leapfrog the data services now offered by some independent common carriers (e.g., Telenet); it might open new attachability opportunities to independent terminal and minicomputer manufacturers, as well as mainframers; it might encourage those firms which have already announced their versions of SNA (DEC, NCR, Burroughs, Honeywell and Sperry Univac) to drop their developments or continue at a cost which would not have to be borne by such competitors who piggyback on AT&T's services; it could open doors for certain firms to develop new, exotic peripherals for use in conjunction with the new services, and for certain software and hardware companies which might be chosen to swell the ranks of AT&T's suppliers.

Defensive Stance

Given the fact that IBM is its chief competitor, it must be kept in mind that AT&T's new thrust might be considered no more than *defensive*, with IBM's opportunities in communications perhaps far exceeding Bell's in computers. Also, it is possible that IBM's SNA specs may be loosened up to provide levels of universality similar to those characterizing AT&T's service; with the proper specs disclosed, microprocessor chip sets could undoubtedly create compatibility between virtually any non-IBM and IBM systems.

Finally, despite AT&T's resource expansion, it may be a while before its marketing muscle in the data communications area can pretend to challenge IBM's.

A key variable determining AT&T's potential competitive impact may be the final resolution of the Dataspeed 40/4 issue. The arguments over whether this terminal should be considered a data communications or a data processing device (AT&T can't offer the latter, according to the 1956 consent decree) seem to be somewhat applicable to networking services as well.

In this light, it is apparent why AT&T is lobbying hard to expand its monopoly scope. If it fails, there will be residual leverage to *deregulate* some operations (thereby effectively circumventing the consent decree).

By 1978, AT&T claims, more money will be spent on data communications and its attendant services than on DP products and services. In contrast, the domestic data communications business — in terms of installed hardware and services — was worth about $5 billion in 1975 and is estimated to be growing at a rate approaching 20% annually over the next five years.

However, the advances in communications during the past five years or so have primarily been taking place in long-haul communications (satellites, microwave, etc.), whereas the important problems of network reliability and, at another level, high costs, lie chiefly in the area of *local* distribution.

Perhaps fiber-optic cables will resolve this key problem and lift the potential limitations on market growth over the next five to ten years. In this regard, Dr. Howard Frank, President of Network Analysis Corp., stated: "The optical fiber may indeed be AT&T's answer to the projected competition from satellite carriers. It may well be true that these carriers will, in the next ten years, divert substantial traffic from the terrestrial network to the sky. However, in a later period this may return to the earth through an optical fiber transmission system."

Moreover, he predicted that "if the optical fiber promise does materialize, we will have to think about networks very differently from the way we do today. The resource, which is now limited, will be virtually unlimited. Thus, different resource allocation problems and different cost optimization techniques will be required. Those technologies which look most efficient today may in five to ten years be very poor compared to the next level of technology." May I add that this last statement also applies to today's computer architectures and technologies.

Data Communications Trends

The data communications market in the U.S. alone grew in value from about $4 billion in 1973 to $5.5 billion annual revenue in 1975. By 1985, AT&T projects a domestic market of over $22 billion. (In constant 1975 dollars, this represents a compounded growth rate of 15% per year.)

However, the shape of the data communications "revenue pie" is changing dramatically (see Figure 5-5). The change in the relative shape of the pie slices is primarily a result of technological improvements. Clearly, AT&T cannot afford to participate solely in the transmission services aspect of the business, given that segment's expected, relative shrinkage to a 20% (or $4.4 billion) share of the 1985 market configuration.

Today, approximately 2,500, mostly dedicated (private) data communication networks exist in the U.S. This business is shared by many vendors, with no single one of them offering a "total system" solution. The environment is further characterized by the existence of many incompatible terminals, duplicate networks per application and low line, terminal and operator utilization.

By 1978-80, AT&T wants to improve this condition by combining computer and bandwidth services to allow any-terminal-to-any-terminal data communications, any host computer connection, and coexistence of both old and new or "foreign" terminals; and by offering a shared intelligent network, thus making available low-cost credit checking, electronic funds transfer (EFT) services and specialized terminal products.

In assessing the office communications market in the context of the "office of the future," AT&T claims that while a "typical" company 25 years ago was spending 20%-35% of its budget on the office function, today this share has increased to greater than 50%. The five office functions which most readily lend themselves to automation are:

1. The *preparation* function, through page display devices, word processing systems, optical character recognition (OCR), etc.
2. The *control* function, through intelligent (or "high IQ") controllers.
3. The *storage and retrieval* function, using hierarchical storage devices (bubbles).
4. The *distribution* function, through electronic mail.
5. The *reproduction* function, the need for which must diminish since it will be more cost-effective to keep documents in electronic form rather than in hard copy. (If this forecast is accurate, then Xerox's traditional business must change significantly if it is to survive in the 1980s).

These five, currently stand-alone functions will be absorbed into a total, integrated system solution, according to AT&T.

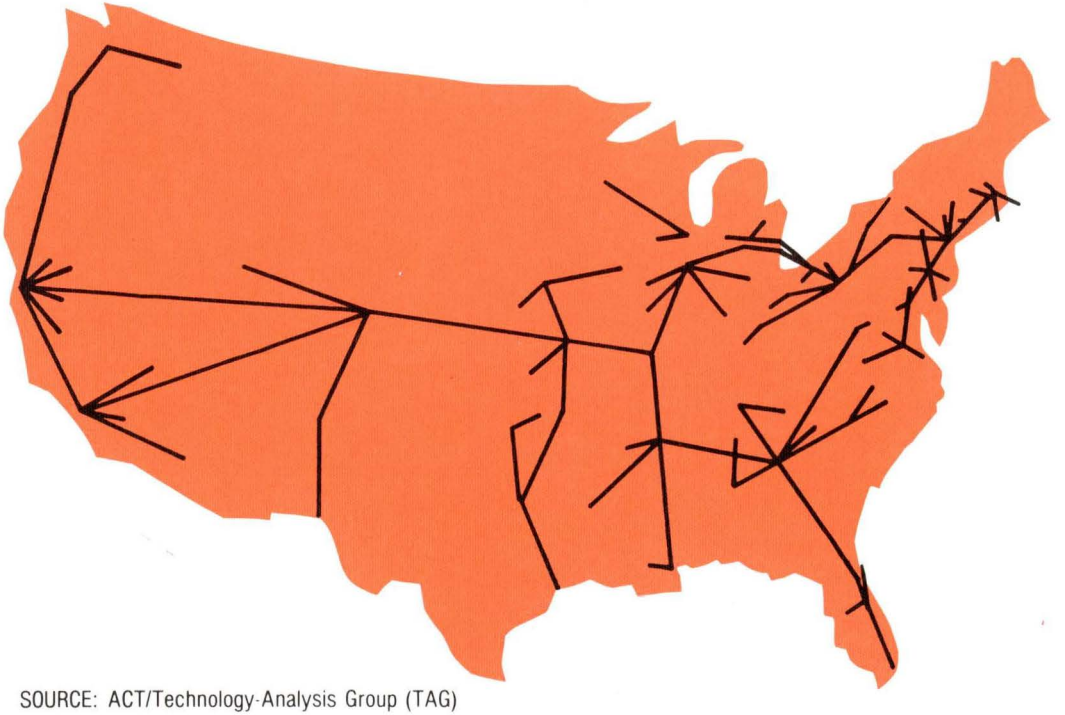
Summary of Critical Issues

The following list spotlights and summarizes those facts which are now (and will likely continue to be) crucial to developments in the data communications and networks area:

- Legal and regulatory issues will be critical to the future growth of the industry.
- Programmable interactive and batch terminals will be in great demand.
- Most computer systems will have at least one communications line by 1980-81.

- Growth in network processors (i.e., multiplexors, line concentrators, nodal processors, front-end processors) will proceed at an annual rate of nearly 50% (starting from a relatively low base).
- AT&T and the foreign Postal, Telephone and Telegraph (PTT) authorities will continue to clash with IBM over protocol standards.
- IBM will not offer distributed control types of networks. Centralized IBM host CPU control and dependency will continue to prevent or reduce independent vendor replacement opportunities.
- Data security and control devices (encryption, etc.) will be essential in some industries and applications.
- Digital microwave and satellite networks will offer improvements, to several orders of magnitude, in line speed, reliability and availability, at lower cost (or at least holding the cost trend down).
- The SBS joint venture will yield in excess of $1 billion in annual revenues by 1983 (assuming that no major FCC, technical or legal barriers arise).
- The IBM vs. AT&T battle will heat up during the 1978-80 period, and the outcome of this confrontation could dramatically change the scope and shape of the data processing market in the 1980s.
- The future of the postal service will in large measure be shaped by the result of the intensifying competition between IBM and AT&T.

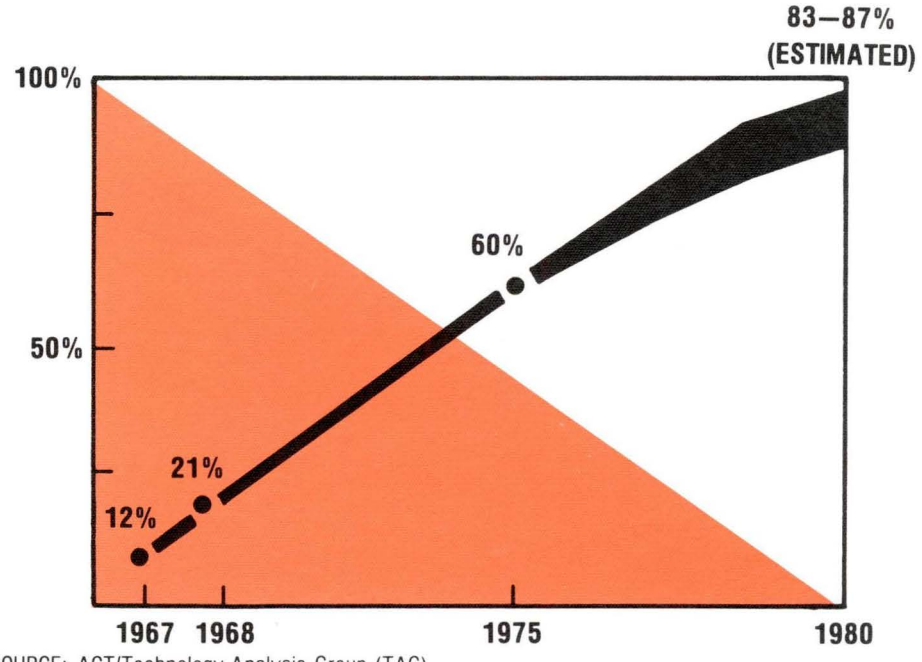
AT&T DDS Cities
Originally Projected For Service in 1976



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 5-1

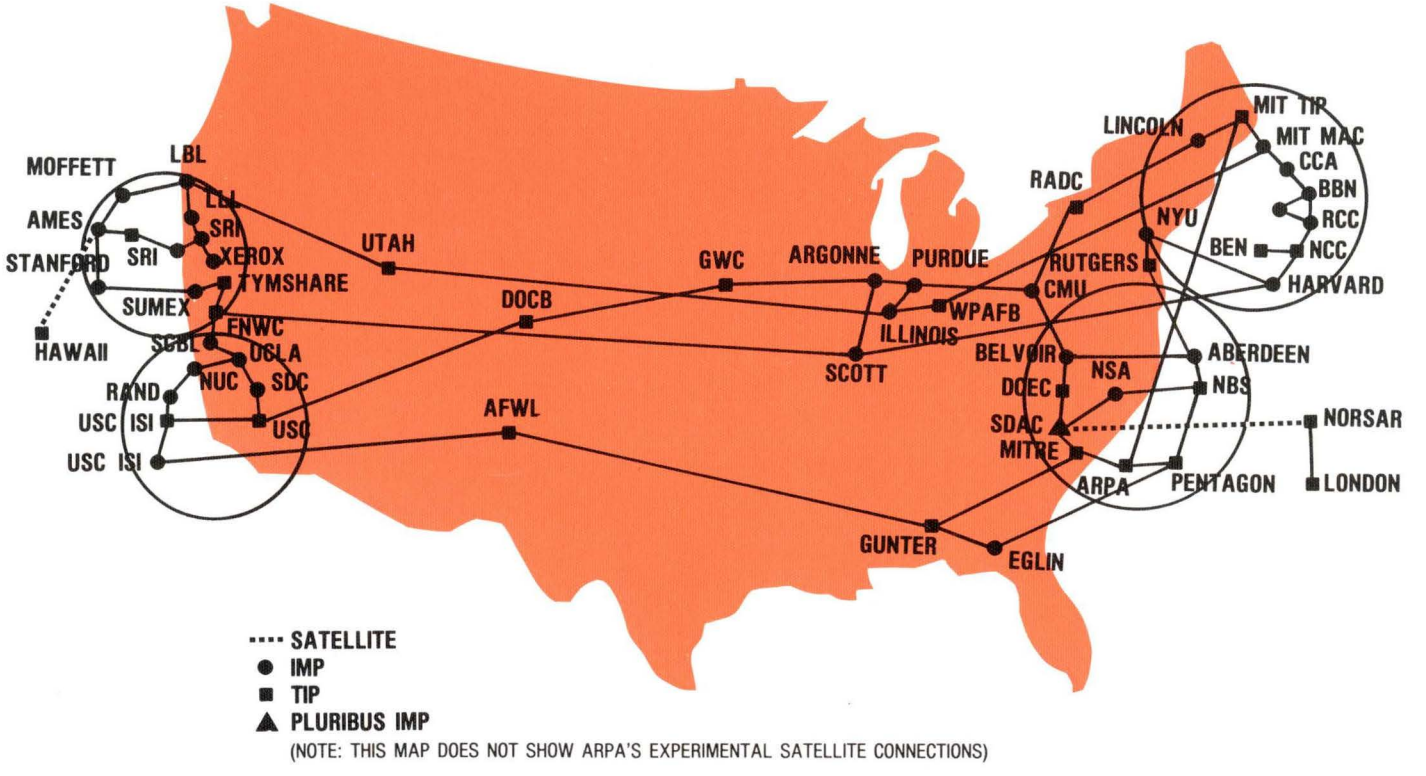
Communications Technology Trends
Data Communications-Oriented Systems



SOURCE: ACT/Technology-Analysis Group (TAG)

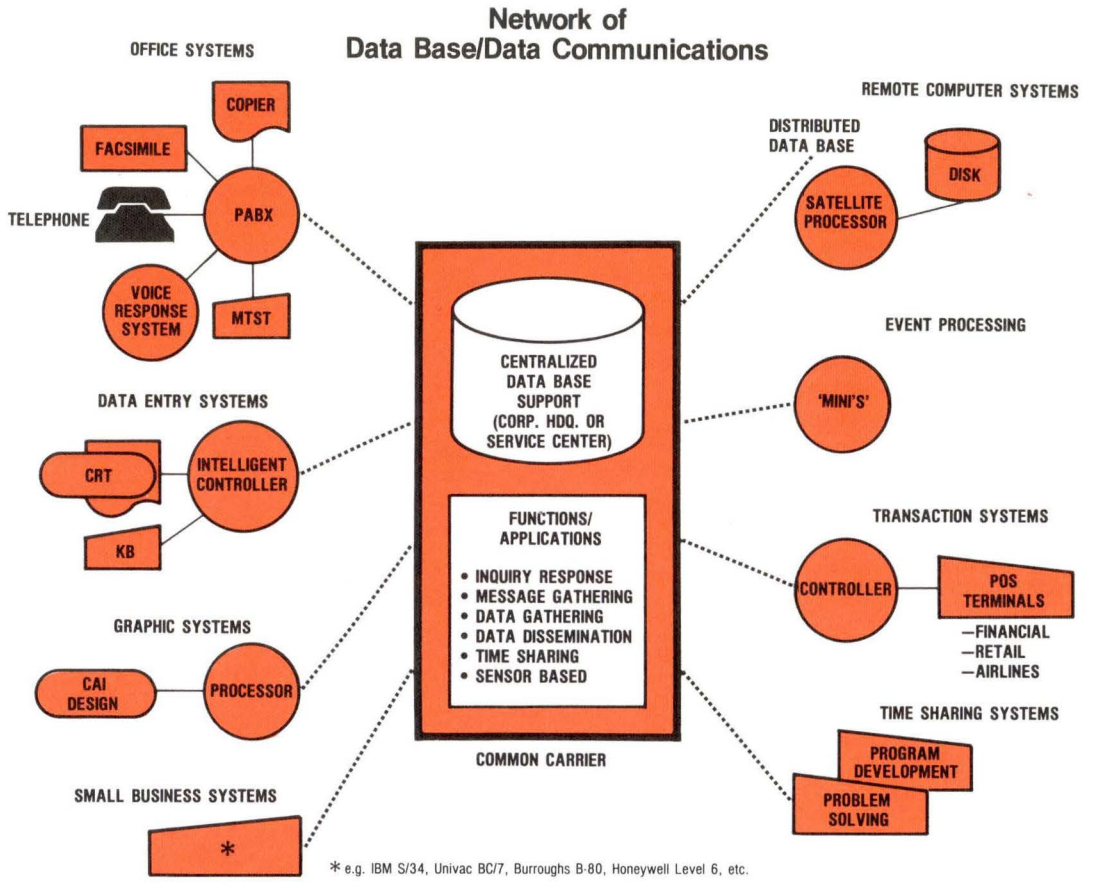
Figure 5-2

Arpanet Geographic Map, February 1976



SOURCE: ACT/Technology-Analysis Group (TAG)

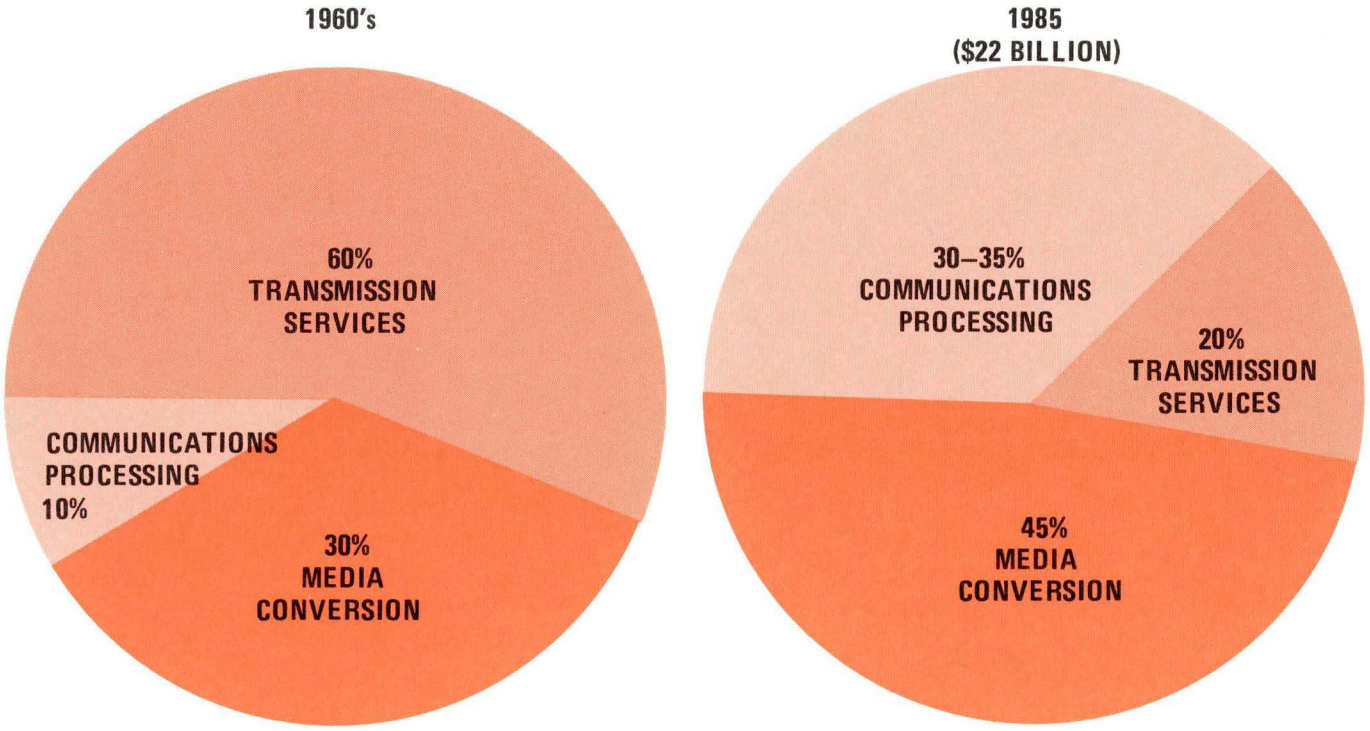
Figure 5-3



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 5-4

The Changing
Data Communications Dollar Distribution



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 5-5

6. DISTRIBUTED PROCESSING

Like birds flocking from one feeding area to another, today's computer technologists seem to be revisiting an old territory with new hopes for nourishment success. Thus, in the latter 1970s we are once again revisiting the field of decentralized DP.

For as long as I can remember, no one would argue that a distribution of DP resources in such a way as to carefully weave these into the operational fabric of quasi-independent corporate entities wasn't the way to go. But few, if any, companies could hope to support their DP needs this way with past technological offerings.

Today's ever-changing hardware, software and communications environments suggest that effective distribution of DP power can be achieved. This could allow for the fulfillment of the most difficult article in Chapter 2's "DP Charter," namely, that "[DP's] success is measured in terms of the proliferation of its capabilities throughout the corporation and the ultimate loss of its own identity as an independent corporate suborganization."

Today, therefore, many corporate entities depending upon centralized corporate DP centers are now lobbying for their own resources in the form of micro- or minicomputers or intelligent terminals. While they are succeeding in many instances, inherent conflicts (among other reasons) with long-term corporate data processing plans are limiting the speed with which they are achieving their goals. We will refer to these entities as end-users to distinguish them from the classic centralized DP organizations established to service them.

Such end-users frequently think they can solve their problems on their own, which at times they can. However, many learn that doing so entails considerable duplication of effort with other end-users or the remaining DP department.

Also rearing its ugly head once again is the de facto long-thought-dead reason for centralization in the first place, that is, the incredibly difficult-to-fulfill requirement that operating entities become involved in matters of technology, including programming, operations, standards and communications. Incompatibility problems, if none others, frequently torpedo any hopes of success.

Many DP organizations, while fully aware of the long-range problems posed by distribution of DP resources, nevertheless accede to today's trend either because of their inability to meet the user departments' growing applications demands, or out of simple frustration at turning current embattled situations around.

Whatever the cause, conflicts between end-user departments and centralized DP organizations are arising everywhere, with substantial casualties occurring among the innocent non-involved.

Today, by the time DP organizations have taken steps to respond to the growing applications development needs of their end-users, individual user departments may have already committed themselves irreversibly to independent, stand-alone or distributed processing solutions. The computer industry's shift towards database and data communications applications further complicates the process of resolving corporate conflicts of this type.

From their own point of view, larger users enjoy certain advantages in this situation which are not shared by small-to-medium-scale users. Their greater resources, plus their access to alternate suppliers, tend to minimize the risks to them of an extremely decentralized approach.

In contrast, the small-to-medium-size users, lacking in both broad-based experience and resources, can easily commit themselves to short-term solutions which will prove in time to have been more apparent than real.

New Class of Competitor

Until recently, the computer sales representative tended to call on and service centralized DP departments. The latter, immersed in the problems of "keeping things going," discouraged or prevented salespeople from interfacing directly with their end-users.

As a consequence of this policy, end-users naturally turned to more responsive minicomputer or independent intelligent terminal salespersons who had no prior involvement with DP center management. As this trend continued, major and centralized mainframe vendors found a new class of competitor emerging that established key links with those people in large corporations who are most involved in expediting and installing *new applications* — end-users. It is these users which control today's key growth areas for the future.

IBM and some of the other major manufacturers are busily engaged in overcoming this problem as they have been known to overcome similar difficulties in the past. IBM's current product offerings in the distributive network systems area include the 3790 (its principal distributed intelligent terminal system), System/32s or the new System/34s, System/3s and, more recently, Series/1, along with the System Network Architecture (SNA) concept and SDLC protocol.

These products and concepts are clearly oriented towards *hierarchical* networks with strong counterdependencies between terminals, controllers, multiplexors, concentrators, front-end communications processors and the central *controlling* host IBM mainframes of which the 30XX models are the latest. IBM appears to be against any approach which implies "distribution of control," where the network components can be easily replaced by competitor equipment.

IBM Strategy

Obviously, IBM, Sperry Univac, Burroughs, Honeywell, NCR, DEC, DG et al will stress their brand of distributed processing, offering greater reliability, more growth potential, a higher level of data security and easier implementation. It is possible that this "sales pitch" has some validity.

In IBM's case, however, on the basis of a review of the Telex trial exhibits and of trends emerging from recent IBM announcements and presentations to user groups, one can see the unfolding of that particular giant's "real" communications and distributed processing strategy. In our judgment, it is composed of the following elements:

1. IBM will protect or lock in all future terminal business wherever possible. The methods employed to achieve this include the creation of host processor dependencies or a "total system" approach where the *full* functionality of the system is not available without IBM-supplied host mainframes, communications processors and their proprietary firmware and software.

   The SDLC protocol, which is unique, also supports this strategy since its full details are not available to other vendors. IBM's emphasis on data security and total network system integrity is also directed at frustrating foreign terminal attachments.

2. IBM will use common building blocks and combine technologies to achieve maximum function-per-dollar, thus greater economy of scale. Some of the methods of implementing this strategy involve the use of "standard universal controllers" (microprocessors) which are adaptable via microcode and software to many different applications.

Another method is the creation of combinations of controllers with specialized I/O components to produce industry-oriented terminals, such as insurance, banking, medical, education, supermarket, retail and factory data collection distributed processing terminal systems.

3. IBM will attempt to provide easier migration paths so that current terminal users may grow to eventually become total systems customers.

   Additionally, wherever possible, the corporation will stimulate add-on business for those customers who currently have IBM host CPUs installed. To implement this strategy, IBM intelligent terminals must be restricted — they will have limited processing capability so as not to impact host systems sales.

   Moreover, the application development languages available at the terminal will continue to have heavy, if possible, but at least some central host dependencies.

4. While in the past IBM has not been prone to playing a leadership role in terminal products, its current continuous stream of these shows it is becoming more and more innovative, creating advanced terminal functions wherever possible.

   For example, plasma displays have been introduced for the financial terminal system (Model 3600), and we expect to see this aesthetically pleasing flat panel display put to even greater use in future products.

   Inkjet printers (Model 40/46) for the Office Products Divison's word processing market represent another innovation. Touch-sensitive screens for very reliable and easy-to-use factory data collection systems are another case in point.

   The more sophisticated, complex, distributed terminal products (such as the IBM 3790, whose price was recently reduced) provide highly integrated features, such as discs, and special buffers that can only be customized or configured at the factory (not in the field).

   The greater use of LSI circuits and microprogramming techniques, the emphasis on reliability, availability, serviceability and replacement modules for easier, low-cost maintenance, and the current "leadership" provided by IBM in the data privacy and security area are other examples of its gradually unfolding product strategy.

5. Wherever possible, the corporation will obviously attempt to minimize impact on revenues derived from existing, older terminal products, such as the 2740/41, 3270 display, 2780, etc. Its current campaign seems to emphasize purchase of these

older products to avoid losses caused by self-impact issues.

Furthermore, one can detect what might be called a "cascade downward" strategy, whereby older products such as the 3270 display are combined with newer products such as the 3790, with such new arrangements intended to help sell the older devices (or, in other words, to unload "old iron"). Confirming this theory is the fact that a new IBM 3270-compatible display terminal was finally introduced on May 18, 1977.

6. While IBM has and will continue to emphasize distributed processing with some host (or multihost) 370 dependency, this does not mean *continuous* host dependency; i.e., the central CPUs should not be allowed to cause the entire network of terminals or distributed processing systems to fail.

   The distribution of processing functions such as data entry, editing, formatting and file updating should allow local operations to continue in a restricted fashion should the host processor(s) become inoperative. A balance will be achieved between total dependency and complete independence from the host CPUs to lock in the customer base without unduly disrupting critical customer operations. (The November 1976 announcement, expanding advanced communications functions to support any IBM terminal network connection to any IBM host 370 [or multiple host] is a further indication of this concern.)

Distributed Processing Trends

Distributed processing can represent a compromise between centralized and decentralized processing. At the beginning of this chapter, I provided a snapshot of some of the chief issues which have lent impetus to the current distributed processing trend.

Others, not necessarily independent, include:

- The issue of human effectiveness versus system efficiency (i.e., hardware performance-per-dollar versus people productivity).
- Political issues, such as the fact that centralized information systems, now a vital part of the headquarters management function in today's decentralized organizational structures, are frustrating control by decentralized management over its own information systems requirements.
- The fact that control over information means greater political power — in Orwell's *1984* the interlocking relationship between history and he who controls the history books is virtually axiomatic.

- The corporation and its organizational subentities are increasingly dependent on data processing systems in such a way as to make availability and reliability critical issues in some industries (e.g., airlines).
- Data processing's growing percentage of total corporate budgets is causing corporations to seek means to optimize their (i.e., DP's) expenses while caring for and feeding this vital resource.
- Technological improvements and developments (e.g., microprocessors, minicomputers and I/O devices) mean that smaller company organizational units can now afford their own computer, or at least sophisticated terminal systems with full processing capabilities.
- Communications costs have *not* dropped as rapidly as other technology performance-per-unit-of-cost areas (for example, micros, minis, etc.). Therefore, remote access to a centralized data processing facility remains costly. (This fact alone in many cases represents one of the major causes or forces encouraging the current distributed processing and mini/micro computer phenomena.)
- Security of DP resources in times when centralized sites are possible targets of dissident groups is a factor in some decisions to distribute the corporation's nerve center and databases.

Search for Definition

It would be foolhardy to suggest that we may provide an all-inclusive definition of distributed processing for reasons which are clear to the majority of the targets of today's purveyors of such systems. Each company marketing distributed data processing (DDP) clearly touts products which are distinguished by both dramatic similarities and differences. Certainly DDP environments seem to have existed since the mid to late '50s in one form or another, if we include such hierarchically organized systems of hardware/software combines as found in the military, government and large (and/or financially substantial) companies.

Star or ring networks of processors linked by automatic communications (however rudimentary) have been commonplace for over ten years now. Timesharing environments, supporting remote batch processors which are at the same time computer systems in their own right, would seem to fulfill today's DDP concept as well.

Could it be that DDP is not a product but a philosophy, the *commonplace* embodiment of which has been waiting for cheaper hardware, cheaper communications and better software to allow for a more productive sharing of these resources?

I say emphatically yes! Cheaper is here; its companion is, in my opinion, awaiting arrival of the "better software" component to grease the skids for widespread usage.

In any event, common to all DDP systems seem to be the notions of:

1. Distribution of *functions* either *locally*, that is, internal to the computer design (e.g., I/O processors, front-end communications processors, etc.) or *remotely* through intelligent terminals which can perform some limited functions.
2. Distribution of *processing*, that is local or remote satellite computers, perhaps with a master-slave relationship between a satellite and the host (with or without their own I/O devices and database).
3. Distribution of *both processing and control*. Examples include the Arpanet, Sperry Univac distributed communications architecture (DCA) and CDC Cybernet systems, where no single host is in total control of the network, and all computers within the network can control their own processes.

Three additional dimensions of distributed processing are operations, management and applications development, in the sense that some decision must be reached as to whether these functions should be centralized, decentralized, or reflect some combination of the two. Solutions will vary from organization to organization, depending on many factors such as the structural nature of the organization — for example, is it multidivisional or not?

Another factor entering into the distributed processing issue is the logical structure of the applications segment(s) or function(s) to be performed: are the tasks or functions mutually dependent, or independent logical functions? For example, inventory control systems, personnel systems, manufacturing control systems, order entry systems and accounts receivable or payable systems may be either totally independent or interdependent to a greater or lesser degree.

The logical relationship of the databases — whether they are integrated or separate databases — represents yet another controlling factor.

Clearly, the distributed processing decision demands a thorough, trade-off analysis which addresses the following issues (among others):

- The cost of accomplishing a desired processing or logical function.
- The time needed to develop and implement the process.
- The maintenance requirements, controls and standards relating to the process.
- The utility or relative priority of the function or process.
- The expected results and risks associated with a purely centralized, decentralized or distributed approach.

The accompanying table (Figure 6-1) lists some potential advantages and disadvantages associated with principally centralized or decentralized approaches. It is important to realize that this "shopping list" will carry greater or lesser weight depending on the industry or organization doing the shopping.

Clearly, one of the major trade-offs has to be in terms of people effectiveness or productivity versus cost/performance or hardware efficiency. A company's decision has to be based not only on quantifiable factors, but also on more subjective, qualitative issues.

However, we can say that of all the constraints and factors involved, hardware represents one of the *least* important, whereas software and management issues are *very* significant, perhaps even paramount.

The IBM View on Distributed Processing

IBM sees distributed processing as potentially the best of both of the worlds of centralized and decentralized DP. If the corporate data processing department and its end-users are viewed as separate entities, then IBM's approach would seem to give end-users more flexibility in defining and designing the systems they need while, at the same time, preserving the concept of overall corporate control over end-user activities.

IBM literature on the topic uses concepts and products (Systems Network Architecture, the 3790 intelligent terminal system and Series/1) which seem to support IBM's view of distributed processing as the best of both worlds. Not all customers share this view, but IBM's total product offering in this area is not yet fully visible.

IBM is not intentionally giving up the attractive minicomputer and distributed data processing market to its competitors (the long-anticipated "Peachtree" minicomputer was recently introduced as the Series/1). Indeed, it would behoove certain independent minicomputer and intelligent terminal vendors to study IBM's strategy and objectives more fully; to do otherwise could seriously compromise the profitability of these independents.

The main difference between IBM's approach to distributed processing and that of some of the other major competitors is that IBM apparently believes in *centralized host control* over the network of distributed systems, whereas other vendors, such as Burroughs, Sperry Univac, CDC and the minicomputer companies, such as DEC, in particular, believe in distributed control within networked systems or "netted systems."

Honeywell's current position appears close to IBM's in this regard at this time but may, in fact, provide an intelligent compromise between

the two. It is important to understand the distinction between these approaches.

IBM clearly is against the "distribution of control" on business principles; that is, it would like to integrate the host computer, the front-end, the network processors and multiplexors, concentrators, remote intelligent controllers or satellite processors and terminals so that each of these components and functions is ultimately dependent upon the IBM central host(s) facility.

This strategy frustrates the attempts of plug-compatible vendors to detach or penetrate the IBM customer base and, at the same time, ensures greater total system or network control, improved data security and system integrity, and (naturally) greater revenues for IBM.

In contrast, most other major vendors cannot afford to be so restrictive.

Figure 6-1 presents a list of competing philosophy attributes which may be used to identify whether a system may be considered distributed or not. Clearly, it's an unclear issue!

Perhaps the final resolution of DDP or non-DDP rests in the paradox that "learning to live in two mutually exclusive yet coexisting realities is a necessary attribute of today's top DP manager."

Centralized

- ECONOMICS OF SCALE: HARDWARE, SOFTWARE, DATA, SPACE AND TALENT SHARING.
- EASIER MANAGEMENT AND CONTROL OF OPERATIONS, STANDARDS, APPLICATION DEVELOPMENT AND DATA BASES.
- GREATER GROWTH AND EXPANSION OF CPU, STORAGE AND I/O DEVICES.
- IMPROVED TOTAL COMPATIBILITY.
- CAPABILITY TO PROCESS LARGE, COMPLEX APPLICATIONS.
- REDUCES DUPLICATION OF EFFORT.
- HIGHER TOTAL SYSTEMS COMMUNICATIONS COSTS.
- LOWER AVAILABILITY/RELIABILITY.
- CONGESTION FACTORS—OVERHEAD RESPONSIVENESS, CONTENTION OVER CONTROL.
- LARGER INCREMENTAL EXPANSION STEPS AND COSTS.
- LESS FLEXIBILITY OR TAILORING TO END USER'S NEEDS.
- GREATER DEPENDENCE ON DP.
- RESTRICTED AND SLOWER ACCESS TO CENTRALIZED FILES.
- HIGHER COST BACKUP OR REDUNDANCY.

Distributed

- LOWER TOTAL SYSTEM COMMUNICATIONS COSTS.
- HIGHER AVAILABILITY/RELIABILITY.
- GREATER RESPONSIVENESS, LOWER OVERHEAD AND CONTENTION.
- SMALLER INCREMENTAL COSTS FOR EXPANSION AND OFFLOADING FROM HOST.
- EASIER TAILORING TO ACTUAL END-USER REQUIREMENTS.
- LESS DIVISIONAL DEPENDENCE ON DP.
- EASIER AND FASTER ACCESSIBILITY TO LOCAL FILES.
- LOWER COST AND MORE EFFECTIVE BACKUP CAPABILITY.
- FLEXIBILITY TO ADAPT TO FUTURE REQUIRMENTS OR CHANGE.
- HIGHER COSTS DUE TO DUPLICATION OF HARDWARE, SOFTWARE, DATA, SPACE AND PEOPLE.
- MORE DIFFICULT MANAGEMENT AND CONTROL OF OPERATIONS, STANDARDS, APPLICATION OF DEVELOPMENT AND DATA BASES.
- RESTRICTED GROWTH; I.E., CPU POWER, STORAGE CAPACITY AND I/O DEVICE SELECTION.
- POSSIBLE INCOMPATIBILITIES.
- APPLICATION SIZE AND COMPLEXITY RESTRICTIONS.
- DUPLICATION OF INPUT, OUTPUT AND FUNCTIONS.

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 6-1

7. USER BUDGET PRIORITIES—PAST, PRESENT AND FUTURE

While net vendor shipments of EDP systems may have declined in 1975, user spending actually increased, in many instances due in part to pressures generated by the recession to achieve improved productivity, efficiency and management control.

With the recession somewhat abated, 1976 may have topped the peak growth record set in 1974. However, the user spending mix continued to change in 1977, with the most apparent shift being in the continuing decline of computer hardware as a percentage of the overall total user budget.

For example, Figure 7-1B shows user budget structures for the large Fortune 1,000-type customers. It can be seen that, compared to the early 1970s, "people costs" are expected to climb by 8%-10% to a level representing nearly 50% of the users' current budget, while hardware costs grow at a lesser rate, i.e., 6%-7% in 1976.

This offers a dramatic contrast to the estimated 15%-20% increase in expenditures for vendor-supplied application packages, unbundled program products, and other utilities and software.

Finally, miscellaneous costs for such products as remote terminals, supplies, communication lines, multiplexors, concentrators, modems and many other items, grew by 12%-14% in 1976.

No Choice for Users

Recent user surveys also show that many customers (especially IBM's) have no choice but to upgrade and expand their systems because, for example, the medium-scale user is typically running two-and-a-half shifts and CPUs are approaching the 100% utilization level.

For example, one of Advanced Computer Techniques' own installations, a 370/145 using CICS and OS/VS frequently resulted in extremely high CPU activity, even from minimal user input. On the occasions when the system has been operational in a restrictive environment (created solely by default), a single user input resulted in the deactivation of other partitions' tasks as the CPU became totally committed to the processing of that user input. In this instance, where paging reached 30-plus pages per second, CPU utilization reached 35%-45% to handle one transaction.

Alternatives left to users facing this and similar situations are clear: either buy high-priced and not always successful IBM field support services to correct software deficiencies, or upgrade. Many users with five-year plans already in place (and thus not easily alterable) were forced by the exigencies of those plans to upgrade their facilities in spite of temporary recession.

Another factor in the trend to overall data processing budget enlargement is that data processing customers throughout the world (but especially in the U.S.) have apparently been deferring decisions on computer procurements or upgrades for perhaps too long a period; in view of this, and given that delivery lead times typically average 12 to 18 months, the consensus seems to be that this year should prove exceedingly strong for most (but not all) manufacturers.

But, perhaps, a major factor in the increasing DP budget situation derives from a phenomenon whose character compels us to break the longstanding taboo that often drives discussion of shifting white-collar budgetary allocations underground. Chart 7-2 says it all: as time passes in corporations heavily committed to automation, less and less is being performed manually, more and more automatically (i.e., through the use of computer-based systems).

Each heretofore manually conducted, now automated, operation must serve to force the shift of clerical dollars from rooms housing seas of white-collar workers to rooms containing automata. Also, the percentage shift of these funds is occurring at a rate in excess of corporate growth rates and will continue to do so throughout the 1980s.

Perhaps it is time to speak the unspeakable. It appears clear to me that the long-held notion that automation merely *creates* new jobs has proven itself to have been illusory. Dr. Carl Hammer, Sperry Univac's international sage, gave an example of this in a recent *Computerworld* article. He notes the impact of "chip" technology on certain traditional, mechanical industries and predicts that the programmer, too, will feel it as the problems of verifying software are met at the chip level.

A curious phenomenon relating to the shift of white-collar dollars to "steel belted" circuitry has to do with management's perception of

the process. Automation tends to transform white-collar wastage heretofore flowing from seas of worker desks into an apparent high-pressure cash flow out of the computer room. While management was accustomed to the former (almost to the point of obliviousness), the now clearer (albeit lesser) amounts of wastage occurring in the latter engenders in management a certain nervousness which suggests a thermodynamic law analogy.

Thus, with all other economic related things held constant, reducing the floor space on which work is to be performed heats up management and reduces its patience. EDP managers faced with explaining increases in budgetary requirements in the face of this phenomenon are frequently at a loss to do so. But then, we hope the waves of change caused by automation infusion will carry along with their increases in automation efficiencies decreases in management insecurities.

The European Situation

Any evaluation of worldwide data processing budgeting trends must, of course, address the economic situation in Europe. According to "The Computer Industry" letter published by Gideon Gartner of Oppenheimer & Co., projections of real GNP growth rates in Europe in 1976 showed Germany with 5%-6%, France with 5%, United Kingdom with from .5%-2%, and Italy with a rate of from 0%-2%, (figures which do not reflect the probably impact of OPEC's decision to hike oil prices in 1976-77).

Projected changes in consumer prices for 1976 included increases of 5% for Germany, 11% for France, 15%-17% for the United Kingdom and 16%-18% for Italy. The Gartner analysis also suggested that the rise in consumer prices may be taken as a direct reflection of widespread wage inflation in Europe, which persists despite the imposition of wage controls in some countries.

These relatively high inflation rates, when viewed in conjunction with the average 15%-20% *decrease* in price per work unit in computing outside the U.S.*, must clearly have their impact on data processing budgeting in Europe — probably an expansionary one.

According to a Sperry Univac presentation in Paris, the per-capita values of installed computers were $189 in the U.S., $91 in France, $70 in the U.K., $30 in Italy and $28 in Spain. These last two figures

*This dramatic decline does not even reflect the most recent wave of price reductions (or performance/$ increases) which the IBM 30XX models and their competitive responses represent.

may help to explain why Italy and Spain remained such strong data processing markets right through the 1974-75 recession.

Sperry Univac also saw EDP total expenditures in Europe doubling by 1980 to $40 billion. Expressed as a percent of total expenditures, the general purpose computer share would, it was said, decline from 30% to 22% in 1980, with terminals, which in 1975 accounted for a 10% share, increasing to 18%, and communications products increasing from 11% to 13% in 1980.

While over the long term, I expect *centralized* DP equipment expenditures to grow at a far slower rate than overall user spending, possibly even declining after 1980, the *total* DP expenditure (given the latest mini and micro-mini offerings) will not.

Another Look at Priorities

To summarize, people costs, salaries, training, benefits, etc. are growing at an accelerated rate in some countries because of persisting patterns of high inflation. Communications costs, including terminals, modems, multiplexors, concentrators and supplies, are gaining a larger share of customer budgets. Software development, maintenance and service costs are also rising swiftly.

In contrast, the central site hardware share of the user's total EDP expenditures is decreasing at a rapid rate. As a result, some manufacturers may not be participating in the real action of the late '70s and early '80s. As stated in Chapter 4, current and future product priorities (as established under manufacturer R&D budgets) must be reevaluated to determine whether the vendor is properly addressing his potential share of the rapidly growing market and promising product areas. Innovative EDP manufacturers must somehow reduce future personnel, software, maintenance and communications costs by building more productivity functions into hardware, firmware and software.

This also reinforces a point which we made in an earlier chapter; namely, that a MIPS-per-dollar (MIPS meaning millions of instructions per second) strategy is ultimately doomed to failure. For example, some major manufacturers may be focusing their limited R&D resources on building high performance "super-number-crunchers." Consequently, these vendors may not be able to afford to participate in other exciting growth areas as well, such as terminals, communications nodal processors, software and network systems, among other potentially more profitable opportunities.

The growth of planning and development budgets at a steadily ascending rate perhaps reflects the elevated priority that applications

software is assigned in many user operations today. The decline of hardware spending will be due to both to lower unit costs for all hardware, as well as to the greater emphasis on low-end, low-cost equipment (more notably minicomputers and intelligent or microprogrammed terminals).

It might initially be thought that by the mid-1980s this decline will reflect a relative saturation of the market for medium-to-large-scale mainframes or computers as we know them today. However, we should bear in mind that the definition of what we currently refer to as a "medium-to-large-scale" processor will certainly have changed substantially by 1985.

IBM's recent Model 3033 introduction may appear to contradict this trend; i.e., the 3033 offers roughly 1.7 X the 168-3's raw performance at nearly 0.7 X the 168-3's prices or 2.4 X the 168-3's performance/price. Assuming that IBM can ship about 2,000 mainframes (Model 3033s alone) by 1980, this represents greater than 4 MIPS X 2,000 or "greater than 8 BIPS" (billions of instructions per second) capability at that time.

The key question is: will the IBM customer base (158 and 168 users) be able to absorb this "greater than 8 BIPS" capacity with new DB/DC applications by 1980?

Apparently AT&T, in view of its recent letter of intent to order 85 Model 3033 systems, is bent on helping out — a curious but not unexpected phenomenon. Letters of intent are not contracts.

Salary Growth

The steady but relatively small growth of salaries as a percentage of the users' total budgets can be partly explained by increased vendor productivity and support and by offerings from service companies, all designed to reduce user personnel requirements.

This should be more evident or effective for low-end systems (e.g., IBM System/32, Burroughs B80, the Sperry Univac BC line, Honeywell Level 6, etc.). For instance, IBM provides audio self-instructional programs which teach the unskilled operator how to run the System/32; and Sperry Univac's BC/7 offers a programmerless capability called "PIXIE."

However, total data processing employment will continue to increase. Staffing of all user installations in the U.S. was estimated at about 870,000 employees in 1975 (including 200,000 programmers), and is projected to rise to approximately one million by 1980 and 1.2 million in 1985 (with a need for more than 300,000 programmers).

Possible Shortage?

A major question is whether the requirement for experienced analysts, programmers and operators can be met by schools or manufacturers in the 1980-85 period. The shortage of skilled professionals and capable project management-level personnel needed to implement complex data-base/data communications systems may severely inhibit the growth of this market in the mid-1980s.

In the U.S. alone, requirements for DP personnel in paraprofessional categories may reach 630,000 individuals in the eight-year period ending in 1985. This represents the entry of at least 10,000 individuals annually, roughly the equivalent of 1.25% of the college-graduating population each year!

This is a chancy requirement and points to a potentially significant inhibitor of industry growth. Recent statements made by spokesmen for computer school businesses suggest that the disparity between industry's appetite for trained personnel and the ability of their institutions to satisfy it may be corrected by the migration of many workers from other fields to the glamour and income of careers in computers. We, however, do not feel that this can immediately answer the question raised by the growing technical complexity of current and emerging systems.

Furthermore, *existing* staff must be qualitatively upgraded. This is particularly true of systems analysts, whose skills are vital to the implementation of complex, data communications-oriented applications and must therefore be kept up-to-date.

The skills of data processing managers are also still too weak. Computing is still so new that it has little history, and thus lacks a reservoir of experience on which it could draw routinely. For data processing managers, this means growing difficulty in managing large projects, increasingly costly resources and problems of the sort one might expect with temperamental, professional people.

Vendor hardware, software and user relations will also have to be improved to ensure greater manufacturer responsibility for systems failure. In the future such failures will have profound and potentially serious consequences.

Improved contracting policies and practices will be required, as will more realistic user attitudes in regard to planning for the eventuality of system failures. Dick Brandon's latest book on contracting, *Data Processing Contracts: Structure, Contents, and Negotiation**, will see increasing usage.

---

*Van Nostrand Reinhold Co., publishers, 1976. 465 pp.

Overstated Requirement

Chapter 1 suggested that a solution to the shortage of experienced personnel (and other complex issues cited earlier) could be the incorporation of more end-user functions in hardware/firmware, such as ease of operation, and other, labor-intensive activities or functions. An illustration of one recent product that partly achieves this goal is the IBM 3800 laser printer and 3850 mass storage system.

Xerox and Kodak are both trying to capitalize on the burgeoning personnel requirement for people who must think by offering products which can do so with fewer headaches. IBM can justify the sale of the 3850 MSS simply by pointing to the reduction in the number of operators required to mount and dismount the tape drives. Also, the reduction in the number of librarians and clerks needed to handle the thousands of tape reels represents another example of IBM's gaining a larger share of the user's total data processing budget.

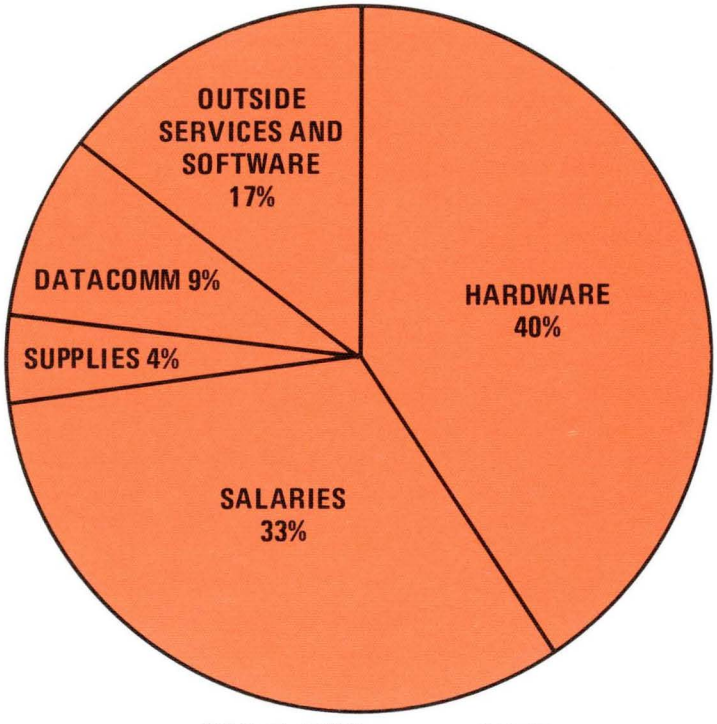
We see this trend of putting more people-oriented functions into hardware growing rapidly in the early 1980 period. Thus, statistics which profile the DP users' requirements for highly trained personnel may be overstated.

The use of line managers, with their business knowledge and experience as data processing analysts, provides another largely untapped source of technical management personnel. The establishment of a wider selection and higher quality curricula in the universities should also be possible.

Finally, a substantial improvement in the use of high-level languages (APL, FORTRAN, macro) and problem-oriented languages in future system architectures, aided by microcode accelerators and other functional and performance enhancements, should ease the personnel shortage problem.

No one can doubt that Logical Machine Corporation's ADAM, in concept if not in fact, heralds a new day in man-machine involvements which will lay RPG, ALGOL, COBOL, PL/1, FORTRAN and the others to rest — and, with these, many user problems.

EDP Spending: Distribution



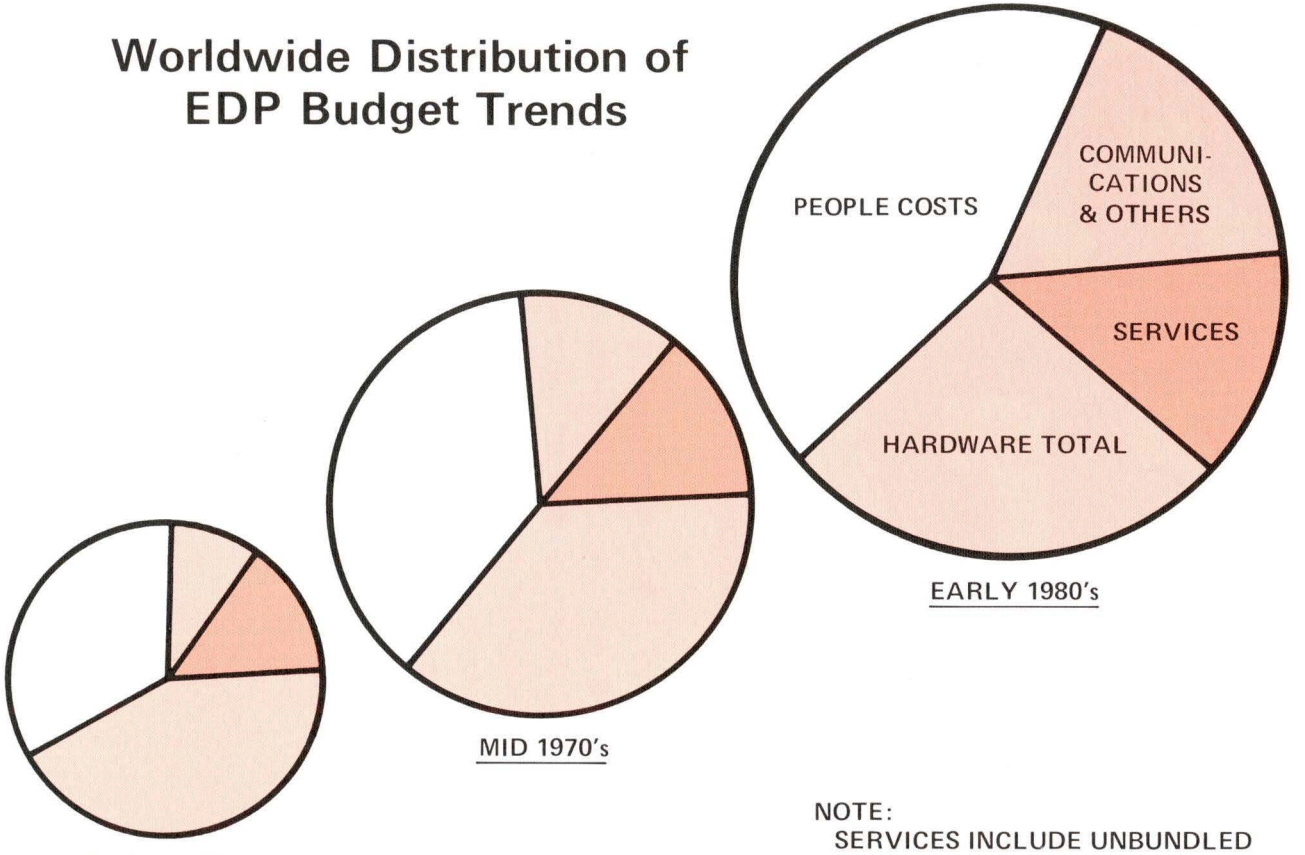
$35.7 Billion — 1977

SOURCE: EDP INDUSTRY REPORT 3/11/77

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 7-1A

Worldwide Distribution of EDP Budget Trends

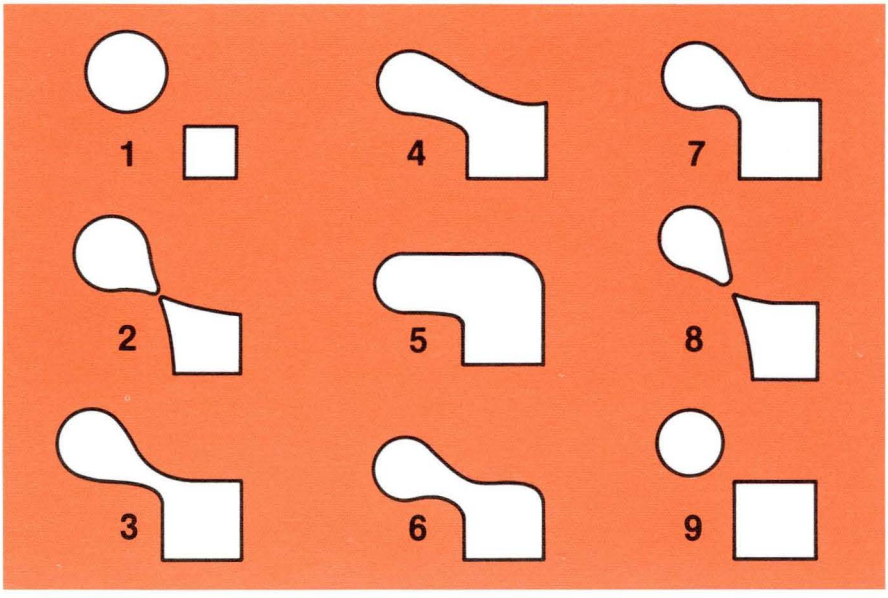


NOTE:
SERVICES INCLUDE UNBUNDLED
SOFTWARE AND OTHER PACKAGES

SOURCE: ACT/TAG estimates based on various industry studies

Figure 7-1B

The Automation Process



NOTES: ● USER ■ COMPUTER FACILITY

- THE ● BUDGET, FOLLOWING AUTOMATION OF AN APPLICATION, SHOULD REDUCE; THE ■ BUDGET SHOULD INCREASE.
- REPEATED APPLICATION OF THIS PROCESS SHOULD IMPLODE THE USER, EXPLODE DP.
- IF THE ● IMPLOSION IS GREATER THAN THE ■ INCREASE, ONE MIGHT CALL THE ROI SUCCESSFUL.

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 7-2

8. FUTURE SYSTEM REQUIREMENTS/EXPECTATIONS

Future data processing hardware is expected to be more specialized and significantly more modular than it is today. "Generalized" local hardware components (common system building blocks) will be *dedicated* (i.e., distributed special-function) for communications, computation, control functions, database management and for other specialized tasks.

These specific task-oriented modules or process engines will be *dynamically combined* according to application processing requirements, interconnected and linked via software and firmware to form the system.

Moreover, remote distributed functions and processing networks (as contrasted with local distribution) will permit various users to control their own local equipment, but a central DP staff must still have responsibility for development, maintenance and coordination of the corporate DP strategy, corporate database and general (interdepartmental) applications software and standards.

Processing will move out closer to the end-user; if one processor or engine fails, the entire organization will not be brought to a standstill.

Systems will be simple and easier to use in this more remotely distributed environment. The user will be able to focus more on the desired logic and solutions than on the processing techniques needed to achieve them.

The major systems suppliers will create a database/data communication environment in which users have greater control of their data and develop their own special uses for the data; however, in the full function of DB management, data must be carefully characterized either as a *corporate* resource or only locally meaningful.

As a Future Requirements Study recently presented by the Guide (IBM 370 Users Group) Future Division put it, "Management is crying

out for DP systems which are more usable. Managers like to manage, and they feel uncomfortable with present day DP systems which are so complex that they defy understanding. . .defy management. . .we can look for users to be more knowledgeable, more demanding and much more involved in specifying the systems from which they derive benefit." New applications should soar when the end-users become more involved.

"User-oriented" systems will require changes in the traditional design of computers. The data management function, or "system resource repository and controller," will become the focal point of the system or network of systems and the processors (CPUs) will play a more "peripheral" role.

"Problem solving" systems will evolve in such a way that hardware and software details or idiosyncrasies become unimportant to end-users.

With high-speed or broad bandwidth data communications available, the system itself will determine how and where processing is done and storage maintained. Operators will make fewer decisions as "secure" operating systems take complete control of the system resources.

Ease of use is perhaps the number one objective. Innovative and complex software, firmware and hardware structures will be required to attain and support this objective. Hardware and software will be strongly interrelated as users become more and more dependent on the systems to assist or guide them in the solution of even the most mundane problems.

Two Network Approaches

Under the network concept, "intelligence" will be available in peripherals and terminals as well as controllers and processors. (Terminals provide multiaccess capability to a computer network. Even a sensor may be considered a terminal, since direct access to a computer is provided.)

Figure 8-1 depicts the current, "traditional" machine organization. All parts of the system are clearly identifiable by the user. In the network systems of the next decade shown in Figure 8-2, the user has access to a limited number of terminal devices. The balance of the hardware and the communications capability becomes essentially transparent to the users. Communications linkages of varying capacities connect the various nodes of the network.

Another approach to a network is shown in Figure 8-3. Here, regional centers control several local centers. Data banks for general purpose use are concentrated only at the regional centers, with broadband communications links between these centers.

The local centers are connected to the regional station via Remote

Job Entry stations and terminals. Local centers thus become satellites to the larger regional operation.

Such an organization of systems/capabilities permits a decentralized system to have some of the advantages of centralized, concentrated databases. It should be emphasized that although remote subsets of databases or special, unique files are available at the terminals, our judgment is that it will be impractical and inefficient to decentralize the *total* corporate integrated database by as early as 1980.

The distribution of total corporate databases will not generally be realistic because communication costs are still too high; large disc files will remain relatively expensive; and concurrent file updating and maintenance is too complex in a distributed DB environment.

In addition, database security and privacy can be more easily implemented and controlled if the sensitive corporate database is distributed over fewer locations, or kept under centralized control.

Future System Architecture

More on-line, communications-oriented, transaction-oriented and real-time systems will evolve as these systems become easier to use, as privacy and security concepts are proven and as systems become more reliable and dependable.

Systems of the 1979-84 period will exhibit such characteristics, although their growth will continue to be restricted by AT&T communications tariffs and line inefficiencies (until AT&T can fully implement digital networks, or until IBM communications satellites and private branch exchanges [or PBXs] become operational).

In short, system architecture in the early 1980s will be more adapted to end-users and problem solving. One can anticipate that IBM's System/80, along with systems developed by other major innovation-minded vendors (Burroughs, Sperry Univac, HIS, NCR, DEC, DG, etc.), will exhibit the following characteristics in support of end-user objectives in the coming decade:

1. *Distributed Special Function Processors:* Instruction stream processors (IP), file processors (FP), media processors (MP), communication processors (CP), sensor or event processors (SP), and a supervisor processor will be provided. Special array or vector processors and essential emulators will also be available.

2. *Interpretive Processing Capability:* For the program development environment, high-level, microprogrammed IPs which execute *source code* (such as PL/1, COBOL, FORTRAN and APL) directly, will be provided. "Standard" native mode IPs will also

be available for the more performance-conscious, production-oriented users. (We already see examples of this interpretive processor or "engine" with the IBM 5100, the APL assist feature for the Model 148, and the Burroughs B7800 FORTRAN processor.)

3. *Multiprocessing:* Multiple *main* IPs will be supported and each process engine will be adapted by Read Only Memories (ROMs), Programmable ROMs (PROMs) or writable control stores to execute a particular high-level language subset or superset. The *non*-interpretive, native mode IPs will execute in-line code at internal speeds which are approximately four times current performance levels.

   Merged transistor logic (MTL), also known as MPT, with propagation delays approaching 5 nanoseconds, will likely be used to achieve these speeds.

4. *Memory Hierarchy:* Most process engines should utilize their own private high-speed CACHE (buffers) as well as up to 256K characters of integrated main memory, and all engines will be able to share a larger (over 100 million characters) secondary or bulk MOS storage. (In this context, it should be noted that "bytes" has been dropped from most IBM 370 documents, and the System/80 "character" may be defined on a dynamic basis, i.e., consisting of 4, 8, 9, . . . *n* bits depending on the function to be performed.)

   Backing up the shared MOS memory will be bubble storage. All four levels of storage (i.e., CACHE, MOS, bulk MOS and bubbles) will transfer data blocks or pages automatically, probably based on a "hardwired," frequency-of-usage replacement algorithm. (See Figure 8-4).

5. *High-Speed, Wide Band Bus:* All the local system resources and process engines will exchange data and communicate with each other via a unique wide band, multiaccess bus. The characteristics of the main bus are uncertain at this time, but IBM and other vendors are known to be working on "fiber optic" I/O techniques which use a solid-state laser source.

6. *Input/Output Interconnection:* A single multiplexed wire (coaxial cable) or loop (dual loops will be optional for redundancy) operating at very high speed will replace today's numerous cables. This loop system will also permit moving more intelligence in-board, in effect replacing the "channel" function with special integrated I/O processors.

IBM System/80 Functional Overview

A conceptual diagram illustrating the probable structure of System/80 with all the distributed "special function" elements is shown in Figure 8-5. (Note that names given to a particular "box" are my own labels. Obviously, IBM's specific model numbers will not be known until announcement time.)

This represents a radical architectural departure from previous general purpose 360 or 370 designs. However, from the standpoint of user software interface, the architecture would appear to be upward-compatible.

The "process engines," e.g., APL, FORTRAN, COBOL and PL/1, are microprogrammed and interpretive; i.e., they can execute source code (except for the native mode, 370 and Fast Fourier processors which represent more tailored or optimized engines).

In our conceptual view, System/80 would function approximately as follows: jobs are submitted locally or remotely via card readers, diskettes, consoles or terminals to the media processor or through the communication processor (and passed on to the media processor) for preprocessing. During this preprocessing, workload queues are created; required system resources are scheduled; data files are requested; and data relationships are established within the Outboard Data Manager (ODM) or "system resources repository," and the particular interpretive processors (for development work) or native mode processors (for production work) are allocated as required.

When all the necessary system resources are queued or scheduled, the job streams or tasks are sent to the Basic System Memory (BSM), i.e., the common, sharable memory, to await execution. As the required special-purpose or native "process engines" become available, tasks are blockloaded from BSM to the faster, private process memory (this memory will probably be ten times faster than the BSM). All required library routines, files and data relationships at this time have already been established and pre-staged onto more efficient BSM or (if necessary) to bubble storage prior to the actual task execution step. At this point the main process engines (COBOL, PL/1, etc.) will execute the tasks to completion.

Should higher priority activities or real-time events interrupt task execution before termination, then the entire task and its status are temporarily swapped back to BSM for later processing and eventual termination.

If the current system workload mix is primarily application development-oriented, e.g., COBOL, then some or all of the process engines could be loaded by the supervisor processor with "COBOL micrologic," which is optimized for program development, testing and/or debug

analysis. (In other words, the system dynamically adapts its resources to the current workload.)

In many situations, tasks will not require any allocation of native or interpretive processors at all; examples include file maintenance, extraction, reporting or inquiry. In this case the ODM, Media Processor or Communication Processor (three functions which, in fact, may be combined in a single "box") will be able to handle such tasks without interrupting the main process engines.

We can find some examples of this "off-line" special-function engine capability today in the IBM 3850 mass storage system control design.

This dynamic adaptation and linkage of resources (logic, control storage and microcode, main memory and I/O devices) to suit the particular process load is substantially different from current-generation general purpose timesharing or multiprogramming concepts. The special-function, locally distributed engines provide dedicated resources for each type of task (this does not preclude reentrant or concurrent use of specific resources) and do not attempt to interleave this task with other different tasks.

For example, a COBOL and FORTRAN compile process would not coexist or have to be multiprogrammed within a single engine.

Another major type of "process engine" may be what I would call the "macro processor" that is optimized for execution of production-oriented, native machine code. After source programs have been fully checked out using the interpretive processor(s), software (compilers) would be optionally available to convert user programs to native object code.

Naturally, many jobs, especially in a university or research environment, are one-shot types which may never need to be converted to more efficient object code. Array processors or Fast Fourier analysis "engines" represent another form of specialized, microcoded processors which execute object code.

Potential Advantages

Some of the potential advantages of the above described System/80 conceptual design include:

- The distribution of supervisory functions — the special processors may simplify operating system control requirements and reduce overhead.
- Better system balance and resource utilization because of the system's automatic "adaptive" characteristics.
- Increased overall performance because of hardware or firmware

(microcode) implementation of many frequently used software functions.

- Improved system security and integrity capability arising from functional isolation.
- Simpler and easier data access, since the system resource repository or ODM handles most data requests and relationships.
- Highly modular design for future expansion and growth; e.g., additional IPs, CPs, ODMs and SPs, can be added as the workload increases in smaller steps.
- More cost-effective redundancy ("fail soft") capability.
- Easier conversion via interpretive processors and emulators.

How will the IBM System/80 be packaged? One possible structure is illustrated in Figure 8-6, where everything is integrated in one total package. The advantages of this packaging include lower reproduction cost; more difficult plug-compatible replacement of individual components; and lower space requirements.

System Software

The System/80 operating system will pull the various hardware, microcode and software elements together into an efficient "information-processing system." The software for System/80 is the key to the reorientation in systems thinking. The major requirement is that it be usable by both professional programmers and management.

According to the IBM trial documents (in the Telex case), "The system must be usable by people who are not programmers, not professionals, not college graduates and not necessarily high school graduates." For the professional, extensions in languages, data handling and system facilities would need to be made.

To meet the requirements of non-programmers, on the other hand, an IBM study group recommended the following:

1. "A layman's language to facilitate system use."
2. "A general purpose query update language (a subset of the layman's language)."
3. "A Common Command Language by which, with equal facility, operators may effectively manage both simple and complex installations."
4. "Language facilities that emphasize simplicity for the user, provide inherent tutorial capabilities (e.g., leading by hand, etc.)."
5. "Facilities to implement occupational or problem-oriented languages. . .based on industry terminology."

Programmer Features

The professional programmer would have new features with which to work, including:

1. "Extensions to current languages to facilitate the implementation of data independence, the identification of parallelism within programs, the variable binding of programs to system resources and [the provision of] data flexibility to meet [future user] requirements."
2. "Expansion of compilers to provide interactive execution for computer-assisted problem solving, program development and debugging."
3. "The introduction of all new functions via the primary higher level languages of the marketplace."
4. "Minimization of the requirement to use lower level languages for new functions."

What does all this really mean? The basic thesis is ease of use, and support for the concept comes from the necessary program structures. The end result is that the program restrictions of the past will be reduced or eliminated.

Some additional system software goals are outlined below:

- User orientation
  - — Non-programmers, executives, non-professionals
  - — Bridge gap between man and machine
  - — Unify all system software
  - — Reduce maintenance and support required
- Availability
  - — Isolate users from each other
  - — Correct minor system anomalies
- Opacity
  - — Divorced from hardware concerns
- Greater user security and privacy
- High-level languages only
- Migration and coexistence of new and old
  - — Eliminate forced conversion
- Higher level of Reliability, Availability and Serviceability (RAS) "like a utility"
  - — Failure detection, isolation and bypass
  - — Automatic restart and journaling
  - — Automatic reconfiguration
- Multiprocessing support
- Computer networks support

Virtual Storage and Virtual Machines

When virtual storage (actually a design compromise) was first developed in 1956 (through the University of Manchester's Atlas computer project), the primary objective was to overcome the high cost of real main memory. In the mid-1970s, semiconductor main memory became relatively inexpensive with prices dropping steadily.

Is the virtual storage technique really necessary later in this decade?

From the cost-of-memory point of view, the answer is still yes. Moreover, as a programming aid, virtual storage (VS) and virtual machine (VM) techniques are very important. Although MOS storage will continue to decrease in cost, users will never satisfy their appetite for more and more on-line storage.

The IBM 370 virtual storage implementation is crude in that all I/O operations (Channel Control Words [CCWs] and buffer areas) are translated from virtual-to-real-to-virtual by software (no I/O DAT, or Dynamic Address Translation, hardware feature exists). This causes on average about 15% CPU overhead. Actual overhead depends on the number of CCWs, among other factors.

In contrast, the Fujitsu M-190 design, which is 370-compatible, has already implemented an "I/O DAT box."

Virtual Machine/370 is the eighth operating system within the 370 series. This is a technique which allows the system to control multiple "virtual machines," each with its own user operating system.

The VM/370 operating system (with microcode assist features) is the foundation upon which IBM will build a massive programming aid capability. A virtual machine concept permits several users to utilize a subset of hardware resources with total separation and protection (while the resources appear to each user to be unshared with anyone else).

With VM/370, the programmer assumes the memory and other resources to be practically limitless (16 megabytes per user, maximum). As such, large, complex programs can be written more quickly. System/80 will create the basic architectural foundation for the new application development environment and for later improvements in software, microcode and hardware technology.

Virtual techniques will also enable IBM and other innovative DP manufacturers to construct information security systems. Privacy of data will be a vital part of the system. For example, control of fetch commands is an inherent part of VS. Critical data will be fetched only when proper authorization is included in the command. In remotely or locally distributed systems, certain processes and data will simply be unavailable to some users.

New and Better Tools

Virtual storage, virtual processors, virtual I/O concepts and virtual terminals will create the framework for achieving the critical ease-of-use objectives. (In this context, the term "virtual" may be equated with "logical" devices.)

In addition to the programmer aids mentioned above, system software should also contain improved debugging tools. Very importantly, system changes should be nearly transparent to the user (system resources should be accessed "logically," not physically) so that previously checked-out programs will be unaffected by hardware configuration changes (except, of course, in terms of performance or response time). This concept is extremely important in that it simplifies upgrading, or migration to new hardware.

Accurate predictions of performance of a given configuration in a given environment must be possible. With this in mind, installation managers will be given new and better tools for evaluation purposes, to assist them in optimizing the total system design.

System profiles, component performance and batch/real-time processing efficiency will all be measured with appropriate new techniques. Use of these methods will even permit the measurement of the impact of one operating environment on another, when both utilize and share the critical resources.

After a system is installed, it should have the ability to monitor its own performance. Workload status as well as hardware utilization information can all be displayed. The total system should remain responsive to changes in conditions, automatically and at all times.

The systems of the next decade must minimize the need for human intervention, and dynamically allocate resources and schedule workloads so that throughput and response time requirements are satisfied. Automatic error correction, retry techniques and on-line failure analysis must be available.

On-line systems of the 1980s will rarely shut down. Continuous processing results in more work output per dollar of investment in the system or network. IBM discussed this "service-free system" (SFS) in the Management Review Committee Minutes of April 19, 1971.

Business Considerations — 1980s

IBM cannot obsolete System/370 with System/80 on a price/performance basis *alone.* Leasing companies, together with independent, plug-compatible manufacturers (PCMs), will find many ways to enhance and extend the life of their vast inventory of used 370s (as they did in the past with 360s).

Newer peripherals, higher density memories and similar embellishments, together with greater discounting, will all inject new life into the "old" partly or fully depreciated 370 systems.

IBM is well aware of this problem and System/80 must, therefore, obsolete 370 on a functional and reliability basis, clearly demonstrating the "ease of usage" or "people productivity" benefits which will more than compensate for reduced prices of "old" 370s from third parties. Obviously, greater "raw" performance/price will also be expected by then.

According to IBM, recent studies of the DP budget have shown "that a larger and larger portion is comprised of personnel cost, as hardware price/performance continues to improve, decreasing the hardware portion." A 1970 study stated that "the personnel portion was 53% of the budget compared to 35% for hardware and 12% other." In 1973, IBM found that application maintenance absorbed 42% of the people costs while only 52% was allocated to "new application development."

Other surveys, conducted by IBM in 1971, attempted to break down the programmer/analyst time devoted to "conversion" efforts, maintenance and new development. These results showed that, on the average, 17% of the application programmer's or analyst's time is devoted to conversion, 35% is spent on program maintenance and 46% on new application development.

Obviously, maintenance continues to grow as more applications are completed, and it is projected to absorb the lion's share of the user's resources by the latter part of the '70s.

A prime objective of System/80 and comparable systems offered by other vendors, therefore, is to use new hardware and software techniques to increase programmer and end-user productivity in the application maintenance and development areas. One way to achieve this result is to reduce the user dependency on DP "specialists" (analysts, programmers, etc.).

Without these productivity improvements, the user would *not* be able to absorb the increased raw computer performance; and the marketplace would ultimately stagnate.

The IBM Outside-In Strategy

During the interim period, prior to System/80, IBM is concentrating on data communications and networks (e.g., recent developments including SNA, SDLC, 3790, 3600, 3650, 3767, 3770, VTAM and NCP) and data management (e.g., 3850, IMS/VS and VSAM). In other words, it is

focusing its attention on the "bridging" devices (peripherals, software) and "appliances" (terminals).

I suggest the reasons for this are as follows:

1. Mainframe purchase activity (e.g., 158, 168 and 30XX) should decrease substantially in 1979-80, and IBM must attempt to fill this potential "revenue gap" with other products.
2. The corporation must get the "appliances" installed before it announces the new System/80 "dynamo"; otherwise, most users will *not* be able to absorb the significantly greater power of System/80 and the market will stagnate (a similar situation arose at the beginning of the 370 era in 1970-71).
3. In order to convert 370 customers over to System/80 as painlessly as possible, they must start a "preconversion" activity *now* (i.e., SNA, SDLC, VTAM, NCP and VSAM are some of the key software/firmware "bridging" vehicles).

The end result (if the strategy is successful) is that many of the remote (external to mainframe) terminal devices will be in place; the important data communications networks and data management functions will already have been converted, and remain essentially compatible with System/80 in advance of the "dynamo" announcement.

IBM's 30XX Processor — A Short or Long Life?

Volume production and deliveries of the 30XX models will not get rolling until about 2nd or 3rd quarter 1978. Therefore the vast majority of IBM customers will install this product in 1979.

Considering the huge influx of orders, one can extrapolate an enormous potential customer financial commitment over five years of nearly $25 billion, based on an estimated maximum shipment of 12,000 units for all 30XX models. This forecast may be broken down as follows: 1,500 Model 3033s; 3,500 Model 3032s; and 7,000 Model 3031s.

In light of this, one must ask: What is the expected life span of the 30XX?

According to R. T. Fertig, V.P. of Advanced Computer Techniques Corp.'s Technology-Analysis Group (TAG), the IBM 3033 and its forthcoming additional members — Models 3032, 3031 and 3033 multiprocessors — appear to represent *interim* products with a relatively short competitive life span. Some of the evidence supporting this conclusion is as follows:

1. Lack of Architectural Improvements

   a. The 30XX (all models included) has a limited address range of

24 bits (16 megabytes) which is insufficient considering the 100 megabyte main storage capacities expected in the early 1980s.

b. Like the other members of the 370 series (and other competitor systems), the 30XX cannot be "certified" as a secure system because of inherent architectural limitations in both hardware and software.

c. The 30XX channel functionality and total bandpass is still too restrictive (1.5 to 3.0 megabytes per second) to handle a hierarchical storage structures with high-speed staging devices such as Charge-Coupled Devices (CCDs), bubbles and bulk MOS storage devices.

d. Although the reliability, availability and serviceability (RAS) characteristics are sufficient for the 1970s, we believe that far greater RAS capability is essential for the critical database/data communications applications of the 1980s.

2. Minor Technology Enhancements

a. The 30XX processor circuits are effectively about twice the density of the Model 168 and not state-of-the-art (e.g., the three-year-old Amdahl 470/V6 or Fujitsu M-190 provide higher density circuits).

b. The 3032 and 3033 processors must be water-cooled because of the high level of heat which must be dissipated.

c. The main storage chips are still 2K bits versus today's delivered 16K-bit memory chips on competing systems.

3. A Bottom-Up Design Change

a. An IBM spokesman explained the approach taken to the 3033 thusly: "We took the best of the 168 design and improved on it" to achieve greater instruction throughput or MIPS; e.g., buffers were expanded, extra registers were added, the I-unit was enhanced, interleaving level was doubled etc.

b. It is also apparent that the 3033 was rushed to the market (believed to be a very short two-year development cycle versus a typical five-to-six-year phased development) in response to serious competitive challenges from the plug-compatible Amdahl, Fujitsu, Hitachi, Itel, CDC and other mainframer products, e.g., Burroughs B7800, Sperry Univac 1100/80, HIS 66/85, etc.

4. A New "Generation" System Should be Introduced by 1980

   a. The current code number* for the System/80 project is "911." The high-end of S/80 (H-Series) is known to be under development by an organization under the direction of Dr. John E. Bertram, IBM Systems Products Division Vice President of Development and Manufacturing.
   b. We believe that "911" will represent a major, but evolutionary and compatible, architectural and technological improvement over the 370 series (including Models 3033, 3032, 3031, etc.) and that S/80 will directly and effectively address the data security, ease of use, RAS and other vital functional requirements for the early 1980s.

5. Accelerating Rate of Technological Improvement (Performance/Price)

   a. In 1971-72, the earlier versions of the 370 series represented less than a 2:1 *mainframe* performance/price improvement over the 360 line (using the Model 50 as a base for comparison); the 360/50 was initially delivered in 1965 and its successor, the Model 145, was first shipped in 1971.

      By 1976, the substantially improved Models 138 and 148 (among other 370 models) established a new performance/price curve of about 3.5 to 4:1 compared to the 360/50.

   b. Finally, by mid-1978 or 1979, the 3033 (also Models 3032, 3031 and MP versions) will nearly double this performance/price (MIPS/$) relationship such that a 3033 processor will offer approximately an 8:1 performance/$ improvement ratio versus the 360 curve.

Dilemma for 30XX

Assuming that the factors detailed in items 1-5 above (admittedly circumstancial evidence) are approximately correct and that some members of S/80 are introduced by 1980, would it be prudent for IBM's 30XX customers to be locked into a four-year lease or purchase contracts in 1979-80? Obviously, the answer to this key question would vary depending on the individual customer's circumstances.

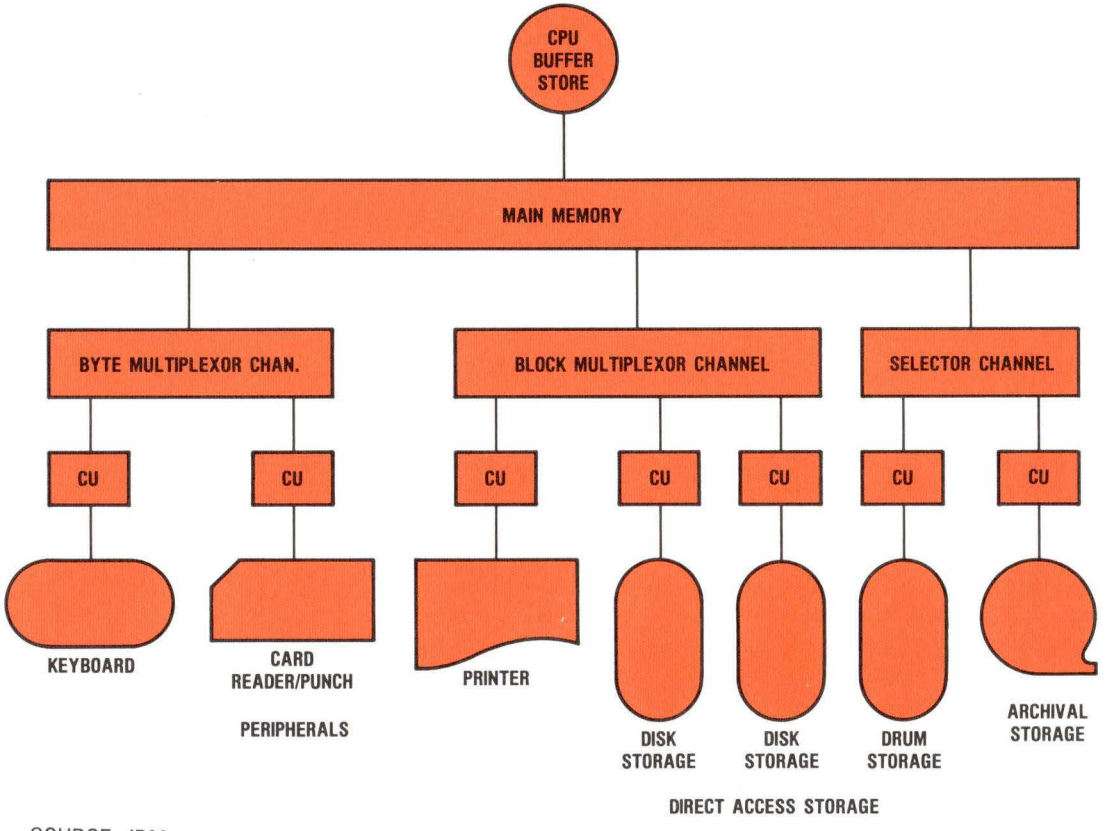
---

*Code names and project numbers change frequently at IBM to maintain security.

IBM management may also decide to extend the life of the 30XX line and delay S/80 one or two years; i.e., why take the risk of introducing new concepts, architecture and technology when the customer base is content with the old structure with improved MIPS/$?

Another possibility is that S/80 may be a bottom-up announcement sequence (S/370 was top-down) and IBM wishes to avoid the typical "bubble" order effect when an architecturally new series is introduced with the obvious initial revenue shortfall. Therefore, one solution would be to lock in the high-end customers prior to unveiling S/80 at the low-end (370/115 and 125 level). (Internally, the low-end models are known as the "E-Series.")

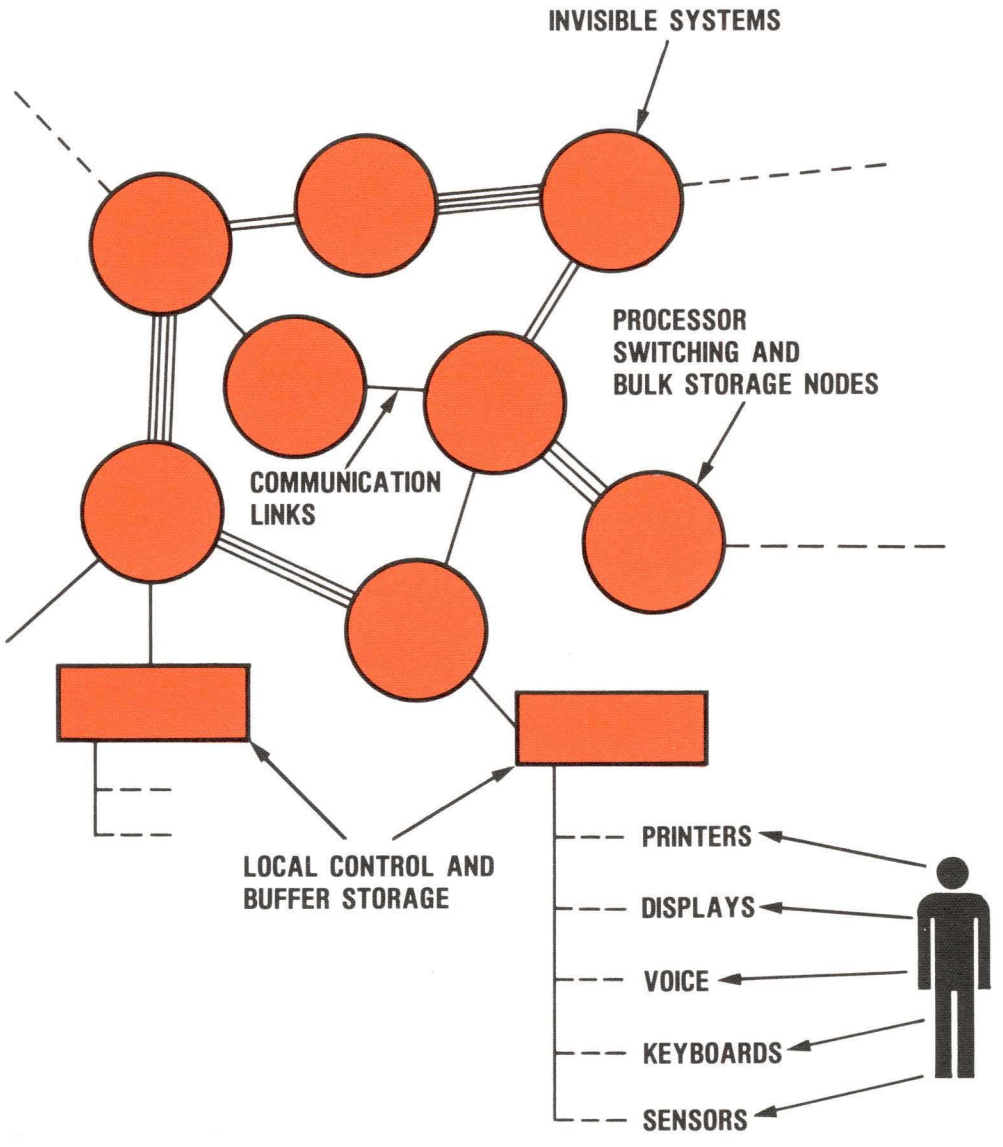
Current Machine Organization



SOURCE: IBM

Figure 8-1

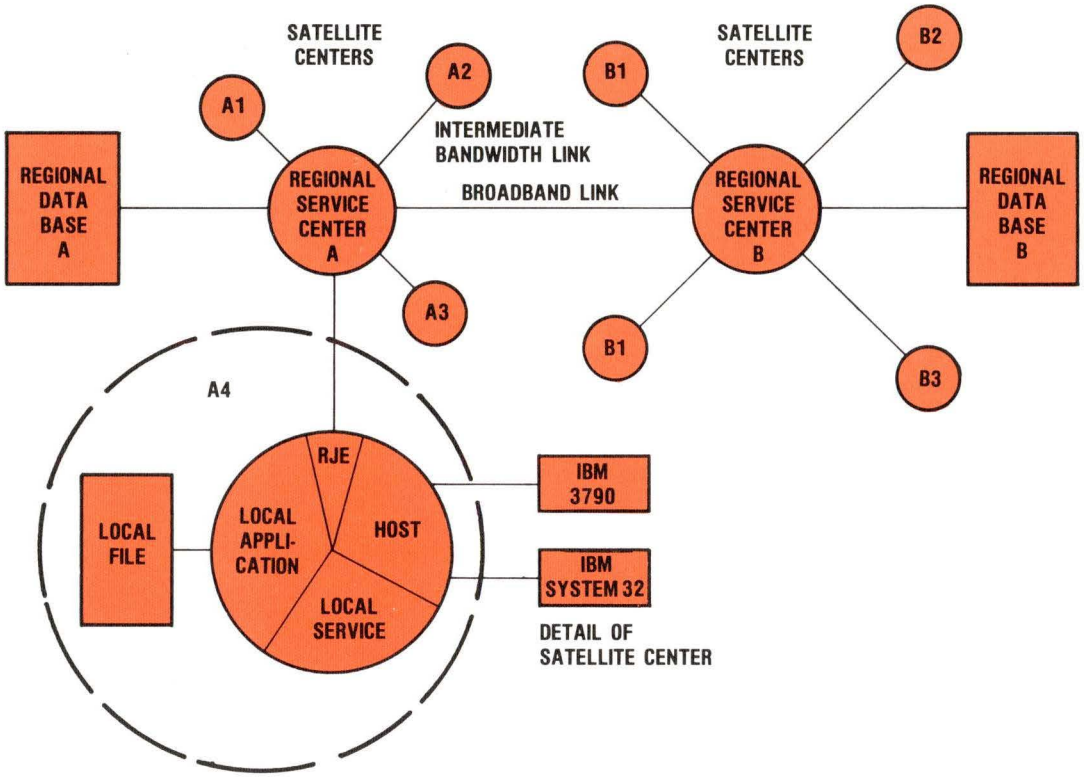
The Network System



SOURCE: IBM

Figure 8-2

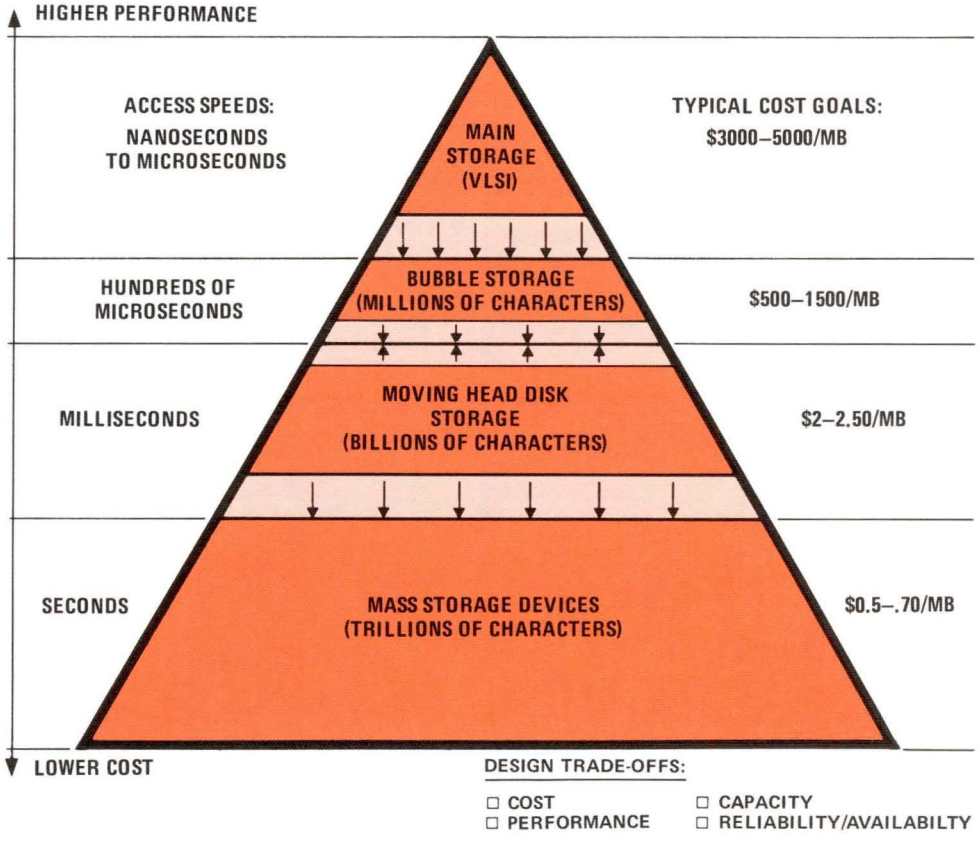
Regional/Local Network



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 8-3

Expected Hierarchical Storage Structures

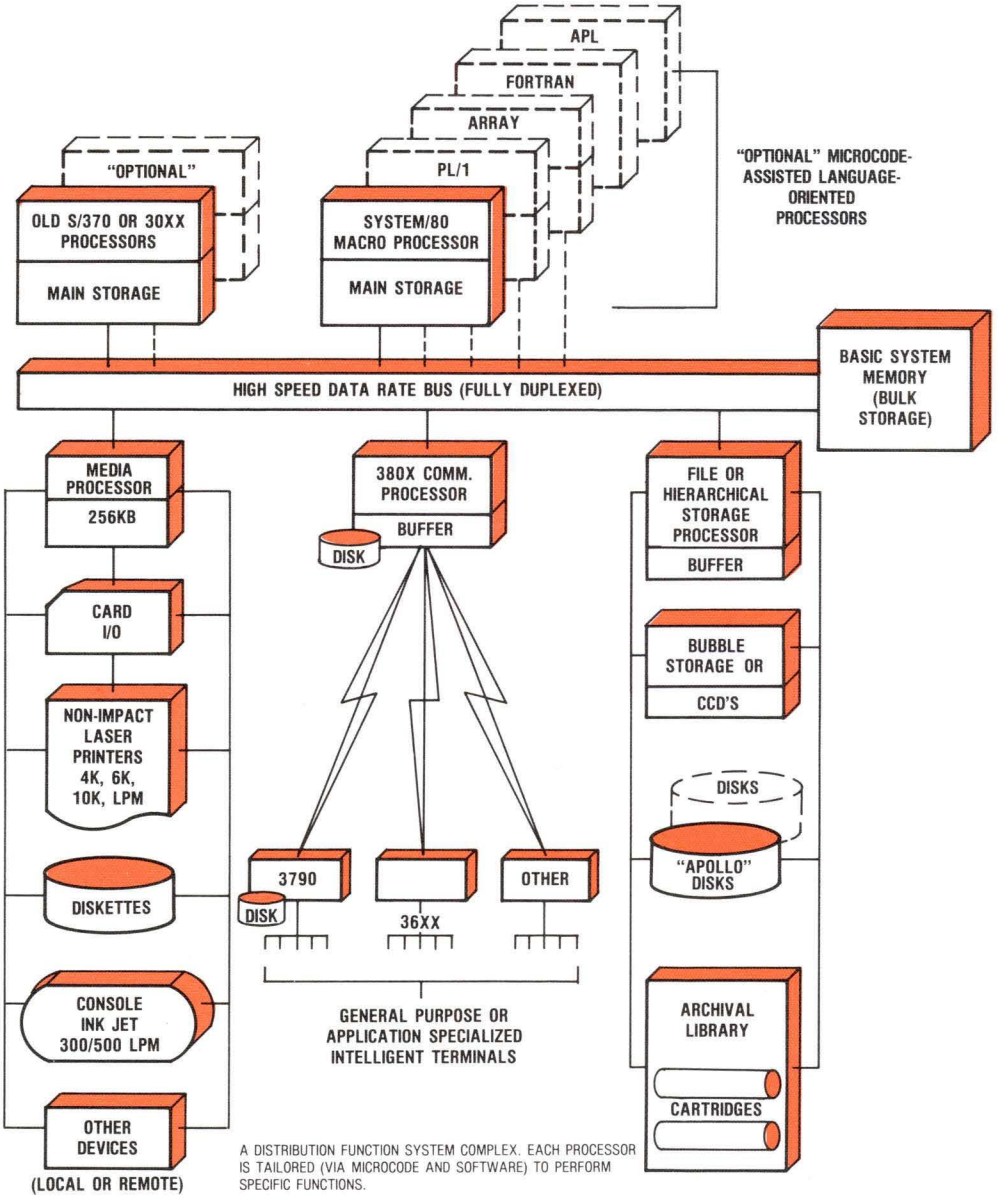


SOURCE: ACT/Technology-Analysis Group (Based on IBM Inputs)

Figure 8-4

A Conceptual View of IBM's System/80

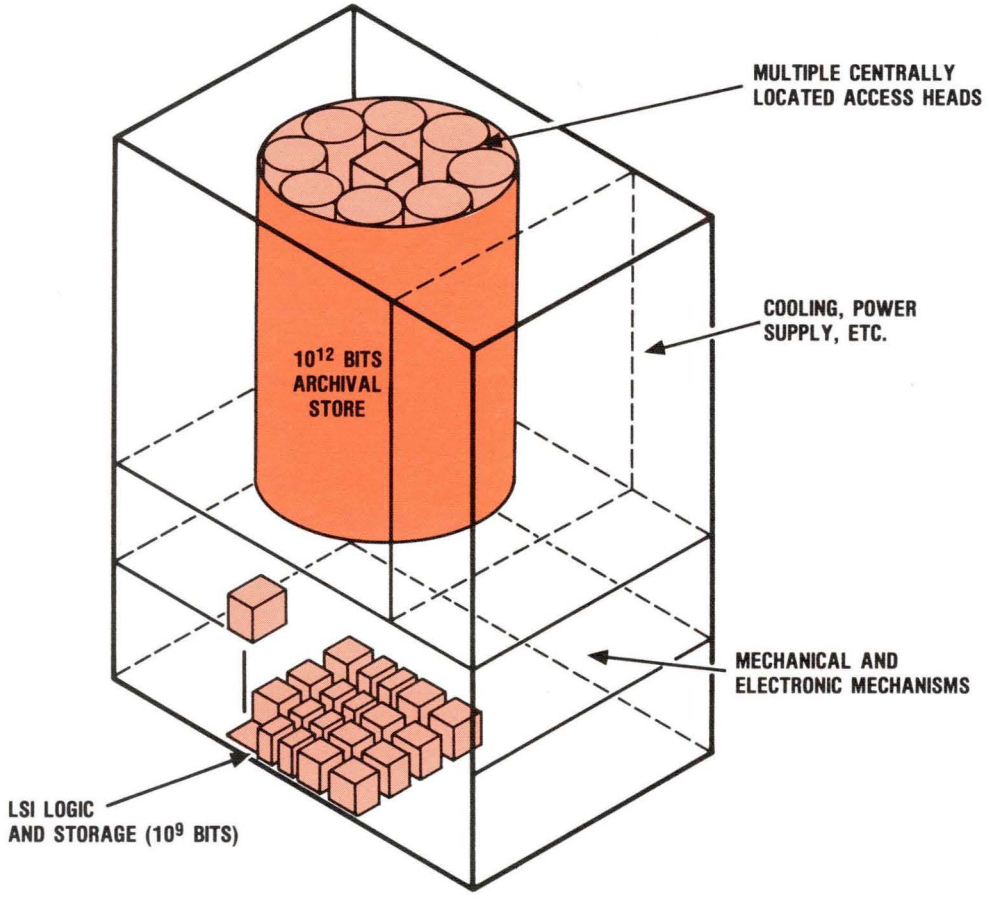
($500K Plus Price Range Only)



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 8-5

A Future Data Processing System
(All in One Integrated Package)


MULTIPLE CENTRALLY LOCATED ACCESS HEADS
COOLING, POWER SUPPLY, ETC.
10^{12} BITS ARCHIVAL STORE
MECHANICAL AND ELECTRONIC MECHANISMS
LSI LOGIC AND STORAGE (10^9 BITS)

SOURCE: Greenblott, B.J. & Hsiao, Muy IBM, AFIPS-NCC 1975

Figure 8-6

IBM's Apparent Long-Range Strategy Summary

Goals	System/80 Methods Available
STIMULATE CUSTOMER PRODUCTIVITY AND GROWTH	GREATER "FUNCTIONALITY," EASE OF USAGE, HIGH-LEVEL LANGUAGES AND PROBLEM-ORIENTED LANGUAGES, RELIABILITY AND AVAILABILITY
UNIFY SYSTEM SOFTWARE AND REDUCE MAINTENANCE COSTS	ONE OPERATING SYSTEM FOR ALL USERS, SECURITY/INTEGRITY AND PRIVACY, DISTRIBUTED FUNCTION, MICROCODE, SIMPLIFIED SOFTWARE INTERFACES
IMPROVE SYSTEM POSTURE VS. COMPETITION AND LOCK IN USER BASE	MORE "FUNCTIONALITY," ACCELERATE TECHNOLOGICAL RATE OF CHANGE, CREATE NEW DE FACTO STANDARDS, INNOVATE
REDUCE PCM EXPOSURES (PERIPHERALS, TERMINALS AND MEMORY)	DISTRIBUTED FUNCTION, INTEGRATED PACKAGING, MICROCODE, MULTIPLE UNIQUE ADAPTERS, SDLC AND LOOP ATTACHMENTS, COMPLEXITY, HOST OPERATING SYSTEM SOFTWARE DEPENDENCY
INCREASE PROFIT MARGINS	EXPLOIT IBM ECONOMY OF SCALE VIA LSI, COMMON CIRCUITS AND BUILDING BLOCKS, UNIFIED SOFTWARE SUPPORT AND MAINTENANCE
TOTALLY OBSOLETE THE 370 (ELIMINATE OR REDUCE USED COMPUTER EXPOSURE)	RAS, FUNCTIONALITY, PRIVACY AND SECURITY, EASE OF USAGE, GREATER EFFECTIVE PERFORMANCE, EASIER CONVERSION, STABILIZE OR REDUCE THE PEOPLE COSTS
OPTIMIZE FE/SE RESOURCES	REMOTE HARDWARE AND SOFTWARE DIAGNOSIS, REMOTE SYSTEMS ENGINEERING APPLICATION DEVELOPMENT SERVICES
PREVENT DOWNGRADING	SPECIAL SYSTEM PACKAGING, SOFTWARE "DIODES," CONTROLLED SUBSYSTEM ATTACHMENT BY SYSTEM/80 MODEL

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 8-7

9. MINI/MICROCOMPUTER IMPACT

In Chapter 1, we noted that today's waves of technological change have obscured our earlier, well-packaged, nicely defined concept of a computer system. Thus it should come as no surprise that there are no universally accepted definitions of the terms "microcomputer" and "minicomputer."

Therefore, it would be well to preface our discussion of the impact of such products by establishing some working definitions which, while they may not satisfy all readers, can at least serve to indicate "boundaries" for each of these product classes. These boundaries, like so many others struggling to maintain their integrity in today's fluid DP environment, will certainly change by the end of this decade: *caveat lector.*

- *Microcomputer* — includes one or more microprocessors which perform logical and arithmetic functions under program control (firmware and/or software), and usually consists of one or more LSI (≥ 100 active gates) chips; also included is some read only memory (ROM) and/or random-access memory (RAM). Today, many microcomputers are limited to 65K or less words of 4-, 8-, 12- or 16-bit size. They provide some I/O interface logic and a limited complement of I/O devices.

  "Typical" prices for microcomputer chip sets *without* I/O devices or power supplies range from under $100 to over $1,000 each, depending on functional capability and memory size characteristics. Examples include the widely accepted Intel 8080 chip, the Texas Instruments TMS 1000 and the Motorola 2901.
- *Minicomputer* — includes a miniprocessor (and may, in addition, incorporate one or more microprocessors). It performs logical, arithmetic and I/O functions under program control at speeds typically five to ten times faster than microprocessors. The mini

memory address range is usually wider than found in microprocessors, with sizes extending from 65K to over 262K words.

The minicomputer price range varies considerably, depending on features included. From a low-end figure of about $10,000, the cost may rise to something considerably in excess of $100,000 for the processor with some ROM/RAM storage and I/O interfaces (but excluding I/O subsystems). In general, the mini price will be under $50,000.

The performance ranges of some minicomputers can overlap those normally thought of as falling in the domain of "large" mainframes, such as the IBM 370/158. In this context, "mini" and "large" assume additional meaning as qualifiers, since the former may match the capabilities of the latter — at much lower prices (in terms of raw MIPS — a somewhat controversial measure).

The market for minicomputers (and microcomputers/microprocessors) has been highly price-elastic. Rapid shipment growth over the past decade has been spurred by an average price decline of 22% since the Digital Equipment Corporation PDP-8 was first shipped in 1965 (Figure 9-1). Technological innovations have enabled minicomputer and microcomputer designers to make processing power cheaper, penetrating markets highly sensitive to price/performance.

By designing more power and functional capability into their units while maintaining price levels characteristic of those belonging to preceding generations, DEC, Data General, Interdata and other minicomputer vendors have been able to penetrate markets for more advanced applications.

Role of Technological Change

Major new micro/minicomputer products have been introduced at (more or less) fifteen-month intervals over the past few years. Average processor and memory prices for these products have declined at about a 20% (or greater) annual rate. However, processor and memory cost constitutes an ever-*decreasing* part of the total computer systems cost; while micro/mini peripherals and software/services costs represent an ever-*increasing* portion of the total cost pie.

For example, the software (utilities, compilers, applications, et al) required by the typical buyer of today's $50,000 minicomputer can involve large multiples of the basic hardware price. In addition, the trend toward more complex systems has nearly offset the price declines of low-end systems, flattening the declining trend in average unit prices (see Figure 9-2).

The fact that market expansion has continued despite the trend towards average unit price stabilization suggests the importance of functional elasticity as an additional factor in the demand equation.

Rapid expansion has also resulted in market fragmentation, with each emerging segment characterized by its own set of multidimensional criteria (industry, function, customer sophistication, etc.). We use the word "characterized" more in the hypothetical than in the practical sense, since a well-defined industrial and/or functional marketplace segmentation which provides a clear basis for product definition continues to elude product planners.

Minicomputer companies have developed families of general purpose processors which were originally targeted for process control-like applications. Moving these into the business arena to do payroll, accounts receivable, etc., hasn't been an easy task. Their entry, at the very onset, represented a horizontal, shotgun approach to the marketplace.

The readiness of end-users in the early 1970s to purchase mini/micro processors is more properly traceable to the economic conditions prevailing at the time (conditions that persuaded end-users to "*make do*" with lower priced, "cheapie" processors) than it is to the perceived sufficiency of those processors to meet more exacting user needs, already provided for by better-accoutered, "classic" products.

As time passed, the market/product mappings improved, causing mini/microprocessors to account for a larger share of the business systems marketplace. Furthermore, the shotgun has now been replaced by the rifle, allowing for vertical penetration increases (Figure 9-3).

Vertical Integration

Partly as a consequence of the trends mentioned above, some of the major mini vendors have launched vertical integration of the manufacture of their own peripherals, terminals, memories and other electronics at a pace well ahead of their competitors. The principal advantages to be gained by this approach include the following:

- Increasing complexity of system configurations (mentioned earlier) means that a greater share of overall system value is represented by peripherals, memory and software (Figure 9-4). In Figure 9-5, note the lower share of system value represented by the Digital Equipment Corporation CPU in the increasingly popular higher value systems (PDP-11/40 and 11/45) versus the traditional PDP-11/05 OEM configuration.
- In-house peripheral and integrated circuit manufacture allows the larger manufacturers to capture a greater share of the total system value-added, a consideration of special importance

to their business in Europe (where independent miniperipheral manufacturers remain relatively non-competitive).

- Improving technology and lower prices have turned low-end CPU manufacture into a low-margin commodity business, with reduced opportunities for product differentiation and premium pricing. Peripherals, memory, software and services have held up eroding profits on low-end products, which are often priced as entry vehicles for capturing subsequent higher margin upgrade businesses.
- A comprehensive line of standard CPU, peripheral and memory components gives some mini vendors the flexibility to construct system packages which more directly address the needs of vertical markets.

In order to successfully carry out the product strategy suggested above, the mini/micro companies need to achieve economies of scale in all product lines through volume shipments. Thus, one or more of the following strategies suggest themselves:

1. Retain and preserve experienced employees while improving productive capacity through inventory building.
2. Form a special components group to help build volume on processors, peripherals and other electronics.
3. Control price concessions and liberal credit terms to lock in higher volume, recurring OEM business. Incidentally, OEM sales by the mini vendors accounted for the lion's share of their revenues in the past; this trend is now being reversed (see Figure 9-3).

With lower costs should come the ability of some manufacturers to sustain market leadership through pricing flexibility and development of additional standard components and configurations (see Figure 9-6). Further cost benefits from higher volume include:

- Economies of distribution and maintenance, made possible by higher geographic installation and shipment densities.
- Additional units over which to spread the fixed expense of developing sophisticated software systems, as well as additional R&D funds to develop applications software for penetrating new markets.

Minicomputer Market Shares

Market shares of the major suppliers are shown in Figure 9-6 and 9-7. According to Bache, Halsey, Stuart, Inc., a respected Wall Street investment brokerage and market analysis firm, "Digital ranks first and will maintain that position indefinitely. Hewlett is currently in second

place, with prime emphasis on larger systems aimed at time-sharing use and for sophisticated industrial automation applications.

"IBM's apparent strong showing is the result of its overpriced and aging System/7, sold into three or four specific narrow markets where massive IBM support can be used to justify the prices.

"Data General's seemingly distant fourth place is deceptive. Data General is the fastest growing mini supplier of size, and is following a strategy similar to Digital's in placing as many systems as possible, laying the groundwork for future larger volume sales.

"General Automation has been growing rapidly, but a series of problems impacted growth in the past year. Honeywell emerges from relative obscurity with its new mini line, which has been quite successful in Honeywell mainframe accounts."

However difficult it is to track the market shares of the larger mainframers (IBM, HIS, Burroughs, Sperry Univac, NCR, CDC, etc.), it is substantially more complex a task to establish the minicomputer market positions with any degree of accuracy. However, it is clear that DEC is, effectively, the "IBM of the minicomputer market" and that, together with IBM, Hewlett-Packard and Data General, it forms the top tier of that market.

Honeywell, General Automation, CDC and others represent the second tier market, although it should be noted that the gap in market shares that separates the two tiers is of formidable width.

Reasons for Demand

According to C. Casale, a Bache, Halsey, Stuart, Inc., Vice President, there are five major factors stimulating minicomputer demand:

1. "Continual declines in semiconductor costs are passed along in minicomputer pricing, expanding existing market opportunities and opening new ones. This trend appears unlikely to change for several years."
2. "Manufacturing processes, office practices and regulatory compliance are increasing in complexity. Apart from other considerations, these spur automation simply to cope with the burdens additional complexity causes. The minicomputer is a primary automation tool independent of process."
3. "Labor costs continue to rise, minicomputer costs continue to fall. Each increment of change adds to the number of situations where automation becomes financially attractive." Consider, in this context, the cost of a given mini as compared to the average cost (in salary alone) to the employing U.S. company of a single clerk — $8,000 to $10,000.

4. "Semiconductor performance gains are being passed on by minicomputer makers as improved access to automation and computation. Each increment widens the number of people that can access computers and reduces the complexities of use. This trend appears to be continuing."
5. "Electronic arbitrage. Mini makers sell cheap what mainframe vendors rent dear. Some leading edge users, such as Citibank, Hartford Insurance and others, are taking advantage of hardware price discrepancies. This trend is in the formative stages since the level of sophistication required to successfully play the arbitrageur role is very high."

The Mini/Micro Impact on the "Mainframers"

The mini/microcomputer or component suppliers will *not,* in my judgment, severely impact the major mainframers for the following reasons:

- When the *system* purchase price exceeds roughly $100,000, the customers generally become more concerned about software, support, maintenance and vendor viability (not raw hardware price/performance).
- The "price war" between the mini/micro and component suppliers will continue to force the price down, and the resulting profit margin erosion will force many smaller firms out of business.
- As minicomputers attempt to penetrate the medium-to-large-scale general purpose markets (IBM 135 and up class, although the definitions of medium-to-large are rapidly changing), the quality and selection of I/O devices are increasingly important.
- Future customers (1980s) will be more and more interested in "system" functionality or ease-of-usage, availability, data security, flexibility, adaptability, operational human factors, application development and maintenance capability — not "raw" MIPS/KOPS/MOPS/Cycle Rates (CPU power) or bits and bytes. Besides, many low-end computer systems are I/O or peripheral limited anyway.

Furthermore, *all* of the major mainframers or systems competitors are gradually replacing hardwired logic modules with microprocessors. In so doing, they are attempting to enhance or expand their current system performance, improve reliability, availability and serviceability, reduce size and power requirements and lower costs.

Figure 9-8 clearly shows the current and projected high annual growth rate for microcomputers (40%). However, this significant growth is from a relatively low shipment base of about $200 million last year. Minicomputers also enjoy a growth rate (25%) which is presently estimated to be about twice mainframe shipments.

However, mainframe shipments may pick up in 1977-78 as the IBM Models 138/148 and 3031/3032/3033 begin to be delivered in volume. Moreover, since large minicomputers tend to overlap so-called mainframes, one must not depend on the absolute values. Figure 9-8 only shows a relative trend over time.

At least a dozen important applications for microprocessors in traditional medium-to-large scale systems can currently be identified:

- Front-end communications processors.
- Remote concentrators/multiplexors.
- Maintenance processors.
- Disc controllers.
- Magnetic tape controllers.
- Hierarchical storage controllers.
- Memory access controllers.
- Nodal processors (in a network).
- Terminal controllers.
- I/O processors (replacements for channels).
- Intelligent terminals.
- Encode/decode (encryption) devices.
- Office automation.

Collectively, the above list represents a very substantial portion of the system's "added value" and revenue. It is important to realize that only two or three microprocessor performance levels are required to handle the above dozen or so functions, and that many systems will require multiple usage of the same type processor(s) (Figure 9-9) for fail-soft operation.

Micro/Miniprocessor Price/Performance Factors

The potential performance capability of a microprocessor or miniprocessor depends on many interrelated and interdependent factors:

- Processor speed is determined by CPU cycle time, instruction efficiency and "richness," word size and efficient memory (ROM/RAM) utilization.
- Input/output capability depends on the number of I/O channels or ports, the channel band-pass or width, etc.
- Memory speed and range (ROM/RAM) can be critical if multiple applications are planned, and speed or capacity range will often represent a key limiting factor in the overall performance equation.

Price represents the most important criterion for many users, especially at the low-end of the microprocessor spectrum (e.g., single chip 4- or 8-bit products such as the Mostek 3870, Intel 8748, Texas

Instruments 9940, TMS 1100, etc.). However, wherever practical, price must be evaluated on the basis of the answers to a wide range of critical questions: for example, how much ROM/RAM is included? are additional supporting circuits required (for I/O functions or power supplies, etc.)? are second sources available? what is the current track record of the vendor in terms of credibility, financial stability, availability of supporting software (tools)? among other less quantifiable considerations.

Summary of 1978-80 Trends

Looking ahead to the near term (1978-1980), there will be:

- Greater competition between IC vendors and mini/microcomputer suppliers.
- Greater competition between mini/microcomputer suppliers and general purpose mainframe manufacturers.
- Increasing consolidation/mergers of IC/mini/micro, miniperipheral and mainframer companies to obtain economy of scale and more added-value (vertical integration).
- Far more emphasis placed on software and end-user support services.
- More cost-effective, larger word size machines (32 bits) and memory ranges.
- Substantially increased usage of mini/microcomputers in distributed systems networks as "nodal" processors or terminal processors.
- Usage of 64K-bit memory chip in products by 1979.

Today's industry-wide preoccupation with distributed systems is heavily influencing product planning in the mini/micro arena. If users are confused with the power and scope of current systems offerings, it appears that the manufacturers are too. The degrees, dimensions and deployment of systems distribution are the key confusion factors. These issues (and others) will no doubt be clarified in the future.

Meanwhile, users are confronted with the dilemma of choosing between multiple mini/micro versus maxi power. Can several midgets successfully compete with one, tall basketball star?

It depends on the court.

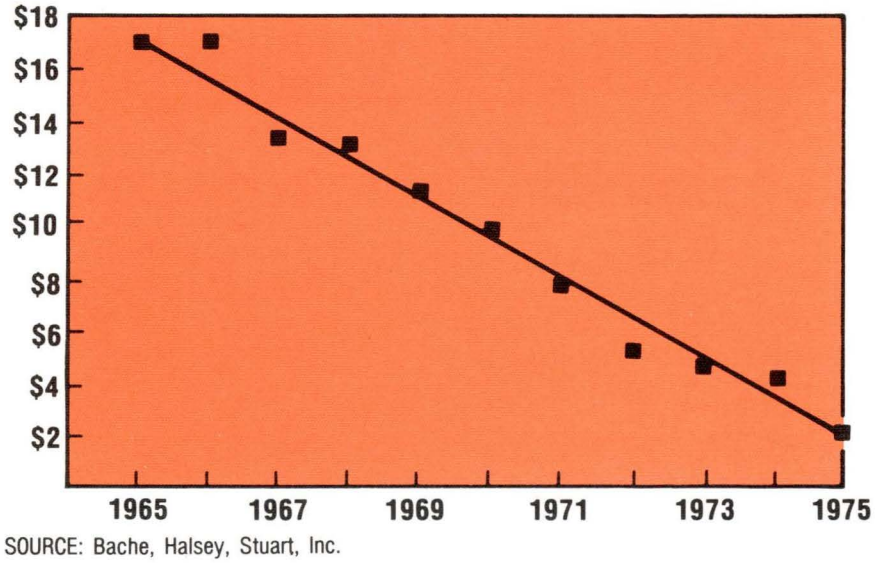
DEC PDP-8 Product History
(12-Bit Word)

MODEL DESIGNATION:	8	8/S	8/I	8/L	8/E	8/F	8/A
FIRST DELIVERY	5/65	9/66	3/68	9/68	3/71	11/72	10/75
MIN. CPU PRICE ($ THOUS.)	18.0	10.0	13.5	10.0	5.0	4.0	2.5
CYCLE TIME (MICROSECONDS)	1.5	1.5	1.5	1.6	1.2	1.2	1.2
TOTAL INSTALLED (JULY 1974)	2,350		6,900		14,950		–

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 9-1

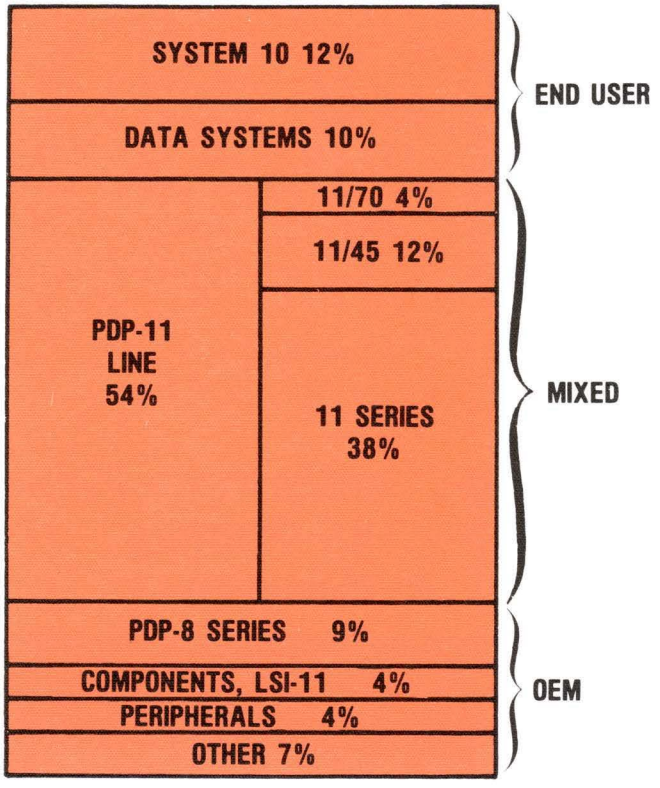
Average Yearly Price PDP-8 With 4K Memory
(In Thousands)



SOURCE: Bache, Halsey, Stuart, Inc.

Figure 9-2

Estimated Breakout of DEC 1976 Shipments
(Total $574 Million)



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 9-3

Estimated Minicomputer (and Microcomputer) Costs Per System

	UNIT SHIPMENTS	VALUE ($ MIL.)	VALUE/ UNIT	PERCENT DECLINE
1967	2,750	$ 220	80.1	—
1968	4,600	272	59.1	26
1969	8,100	355	43.8	26
1970	11,100	415	37.4	15
1971	13,500	377	27.9	25
1972	22,600	500	22.1	21
1973	33,900	770	22.7	3
1974	50,000	1,085	21.7	4
1975	71,000	1,430	20.1	7
1976	96,000	1,820	19.0	5

SOURCE: International Data Corp.

Figure 9-4

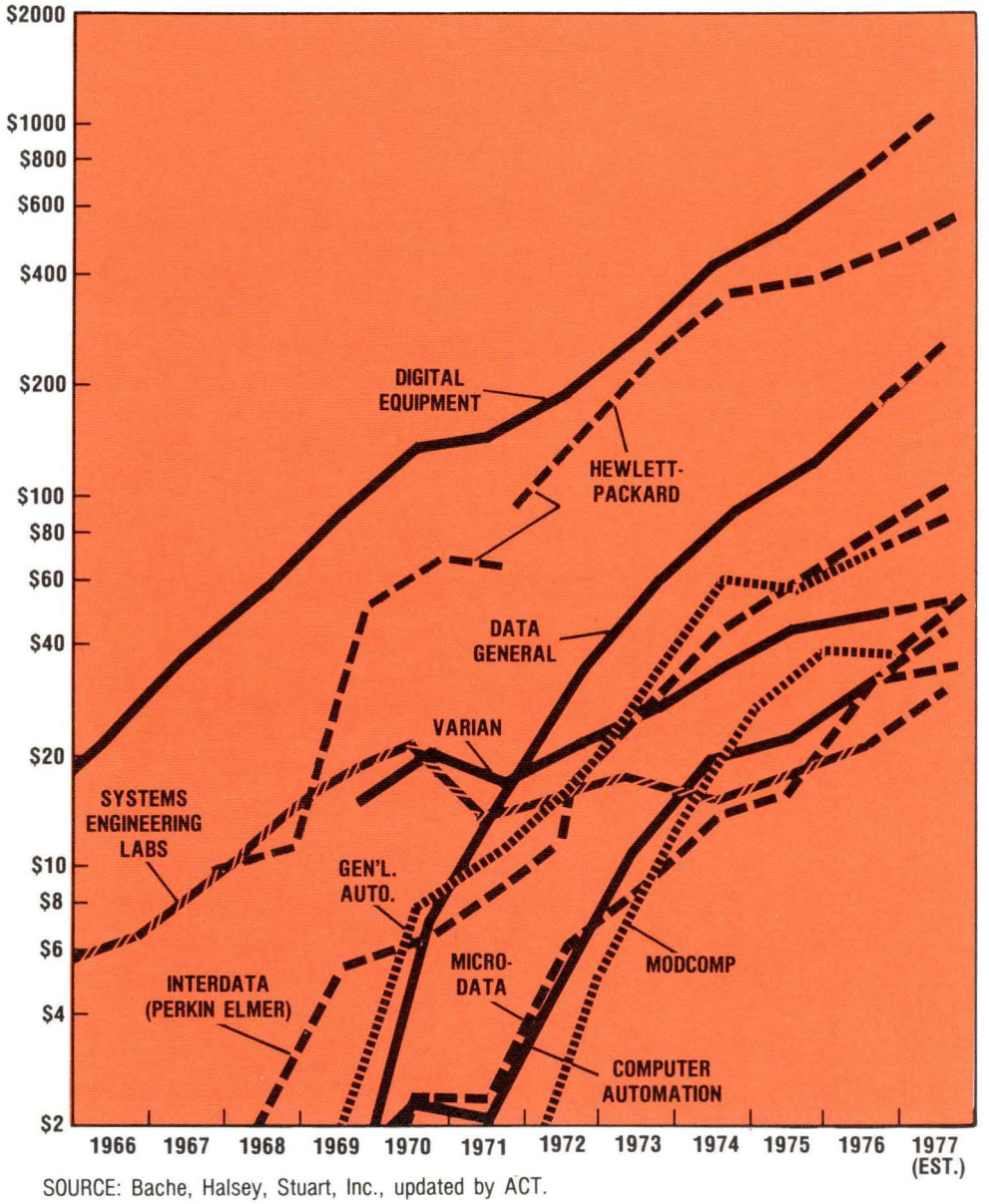
Costs Associated With Various PDP-11 Configurations

	PDP 11/05 OEM CONFIGURATION		PDP 11/40 BATCH CONFIGURATION		PDP 11/40 4 USER TIMESHARING		PDP 11/45 LARGE SCALE SYSTEM	
CPU & MINIMUM MEMORY	$ 6,495	50%	$12,995	28%	$12,995	21%	$ 23,900	12%
FEATURES	320	2	590	1	590	1	5,830	3
MEMORY	—		5,200	11	12,580	21	63,800	31
MAGNETIC STORAGE MEDIA	2,990	23	19,700	42	21,845	36	76,025	37
TERMINALS	3,225	25	3,225	7	12,900	21	3,225	2
CARD I/O	—	—	4,860	10	—	—	2,400	1
PRINTER							30,000	15
TOTAL PERIPHERALS	6,215	48	27,795	60	34,745	57	111,650	54
TOTAL SYSTEM	$13,030	100%	$46,570	100%	$60,910	100%	$205,180	100%

SOURCE: ACT/Technology-Analysis Group (TAG)

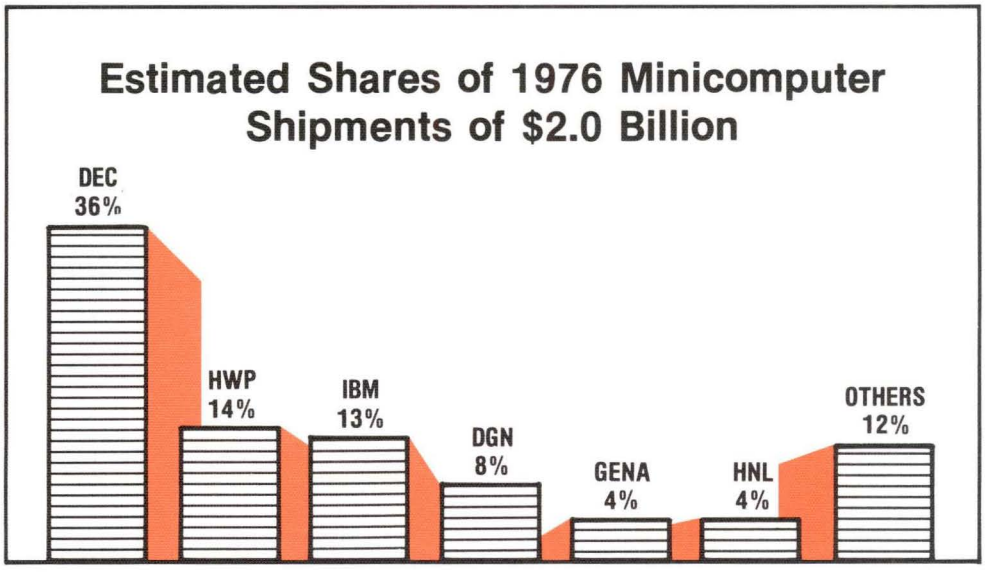
Figure 9-5

Minicomputer Revenues of Selected Companies
(In Millions)



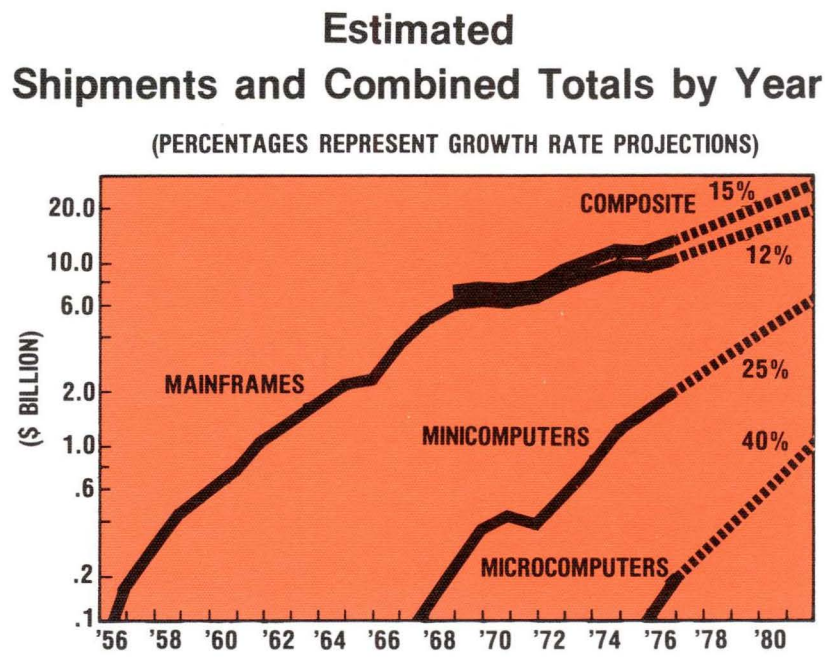
SOURCE: Bache, Halsey, Stuart, Inc., updated by ACT.

Figure 9-6



SOURCE: Bache, Halsey, Stuart, Inc., updated by ACT.

Figure 9-7



SOURCE: Bache, Halsey, Stuart, Inc.

Figure 9-8

Examples of Microprocessor Products

SUPPLIER	MODEL
ADVANCED MICRO DEVICES	2901 AM9080A
AMERICAN MICROSYSTEMS, INC.	S9209 S6800
ELECTRONIC ARRAYS	9002
ELECTRONIC MEMORIES AND MAGNETICS	PC25
FAIRCHILD CAMERA & INSTRUMENT	PPS-25 F-8
GENERAL INSTRUMENT	LP 8000 CP-1600
HARRIS SEMICONDUCTOR	IM6100
INTEL	4004 4040 8008 8008-1 8080 8080A 3000
INTERSIL	IM6100
MONOLITHIC MEMORIES	6701
MOSTEK	5065 F-8
MOTOROLA	M6800 2901
NATIONAL SEMICONDUCTOR	IMP SC/MP PACE
PANAFACOM LTD.	PFL 164
RCA	CDP-1801 CDP-1802
ROCKWELL	PPS-4 PPS-8
SCIENTIFIC MICRO SYSTEMS	SMS 300
SIGNETICS	2659
SOLID STATE SCIENTIFIC	CRD-8
TEXAS INSTRUMENTS	SPD0400 8080 TMS 1000 TMS 9900
WESTERN DIGITAL	CP-1600

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 9-9

10. SOFTWARE TECHNOLOGY TRENDS

As the price of hardware has plummeted, so, too, have the electronic and electromechanical components of that hardware increased in complexity. At the same time, the need for human involvement at the operations level, whether in terms of on-line, manual intervention or elaborate off-line procedural activity, has significantly diminished. Improved software, prepared so as to exploit the capabilities of modern hardware, has to some extent impinged upon and reduced the traditional role of the operator. Indeed, we may well expect the systems of the '80s to run unattended, driven by powerful commands incorporated into applications programs, sometimes making "independent" use of inference or even analogy.

As we shall see in this chapter, the sudden general reversal in relative size of the human and automated cost components of many data center budgets in the '70s has caused us to rethink our priorities in the matter of software creation. Such a reexamination forces us to conclude that the quest to optimize data center costs around hardware operations, frequently at the expense of programmer productivity, has stifled development of the kind of artificial intelligence we truly need to manage current, let alone future, systems. For while striving for operations simplicity, we seem to have forgotten that fundamental to inducing machines to do our bidding is the availability of some *rational* means of telling them what we want done.

The vehicle for this kind of communication should assume the form of a reasonably simple language, one that is quite natural *to us.* Since all languages are unnatural *to machines* anyway, it would seem quite senseless to burden ourselves with unnecessarily elaborate sets of symbols and rules, in whose absence an application cannot be rendered operational, and in whose presence a programmer's sanity is threatened.

Today's most commonly used higher level languages (e.g., COBOL, FORTRAN, PL/1, etc.) are so demanding of programmer precision in coding that, since the early '50s, a relatively insignificant increase in programmer productivity has been achieved through their usage.

Together with inflationary factors, current requirements for learning the operating systems control codes needed to test and run an application seem to play a large part in nullifying the beneficial effects of compiler-level code generating powers. Observers of the DP scene who argue that the (dubious) 3% rise in programmer productivity noted over the last 25 years is related less to the encoding task (estimated to account for 20% of what programmers do) than it is to lack of discipline in systems analysis, documentation, testing, etc. are probably right. But no one would argue that our computer lingos do not affect our approach to conceiving and creating specifications for applications, not to speak of testing and documenting them.

Indeed, it appears to me that the use of the various, current coding languages is, in its own way, responsible (although not perceptibly to any but, perhaps, the skilled linguist) for the siphoning of great sums of money from development budgets in all significant areas of the systems analyst/programmer's endeavor.

The Price of Under-Emphasis on Software

Breakthroughs in chip technology, improvements in materials, new integration of previously independent components, elimination of elaborate connect contraptions, etc., have reduced the physical size of systems manyfold. Clearly, the theory of hardware systems design is on a galloping straightaway in our race to optimize hardware environments. At the same time, however, software technology, while achieving ever greater sophistication in support of efficient hardware operation, has shown little progress in its capacity to facilitate applications specification and preparation. This is, of course, surprising to some industry watchers, since programmers generally use the same old languages to communicate their applications requirements to machines today as they used over a decade ago and so might be assumed, through prolonged intimacy with those languages, to have achieved ever higher levels of ingenuity in their use. Programmers should, it would seem, be getting better.

While it is true that operating system/language processor facilities have, over the last 25 years, evolved to provide increasing power to the user, with this has emerged a concomitant requirement to either increase the technical sophistication of resident software staff or hire

qualified outsiders. Both of these tasks are becoming increasingly difficult to do except at high cost.

The mere fact that we have come to this unfortunate pass serves to support my thesis that a blatant disregard for the special importance of software in the solution of man/machine interface difficulties has existed for a long time, and that excessive concern with hardware efficiency probably spawned it. So, today, we find ourselves in a situation where further advances in hardware operational efficiencies bring decreasing monetary returns, while causing systems analysis/programming costs to soar. If it is true that the origin of this state of affairs may be traced to the days when hardware costs were so high that programmer costs seemed inconsequential, it is equally true that planned software environments haven't as yet reflected the reality of this reversal.

Well, bad habits are hard to break; but in view of what's afoot in our industry even as I write, I can believe that better days are in the offing.

As software specialists scramble to recast their plans in order to capitalize upon the many hardware benefits cast ashore by the waves of technological change occurring in our industry, certain previously hot software ideas cool (and vice versa). Stimulants for this activity (and there are many) include the perennial user/manufacturer concern over increasing shortages of skilled personnel such as are needed to work on systems of increasing complexity. In the early to mid-'50s, software pioneers quite naturally turned to language development as a means to provide users with increased productivity. Later in the same decade, however, the preoccupations of both manufacturers and users with operating systems development impacted continued language development. It is tragic that software designers were not then disposed to view operating system functionalities as implementable through continued evolution of higher level languages. Had it been otherwise, many of today's systems utilization problems could have been avoided. The use of "natural language"-like facilities might even have been commonplace today.

Splitting off the functional particularity of writing what was wanted from the machine from that of determining how the machine was to satisfy demands put to it in such fashion — a seemingly natural separation — sowed the seeds of today's costly confusion over where it is reasonable for our intelligence to end and the machine's to begin. While we may excuse this misdirection as understandable in view of the radical change in personnel/hardware cost ratios between the 1950s and the 1970s, we really ought to have forseen the problem much sooner.

I am not saying that I believe most manufacturers to have been

unaware of or unable to respond to the need to make our systems do more (and hence our people less); but I *am* saddened by the thought that they could not act accordingly. Thus today, many may be faced with seeing their past software investments decline in value more quickly than anticipated. And users, with just as great an investment in past technology, now a competitive weapon in the corporate arsenal, may be caught too.

Whether the reader accepts this picture or not, the struggle to keep user facilities technologically current is siphoning off the skills which might otherwise be put to good use in transforming our still manual (but begging-to-be-automated) applications more quickly, and with greater economy. With the cost of skilled personnel support becoming greater than that of operating hardware — hardware driven by software which has been expressly designed to optimize the hardware's operation at the cost of personnel efficiency — the stage is *indeed* set for a period in which incremental returns on hardware investments cause increasing losses through personnel costs.

It is certain to me that by the mid-1980s, the idiosyncrasies of operating systems will no longer be deemed worthy of our (that is, the users') attention. In my view, that happy state of "benign neglect" is long overdue. But what worries me is that our current language processor facilities, as typified by COBOL, are so inadequate that, pending the arrival and acceptance of the "newspeak" computer lingos of the '80s, we shall have dug ourselves dangerously (not to say irretrievably) deep into the mire of COBOL-originated, COBOL-generated code.

Natural Language and the Limits of COBOL

The hoped-for potential of COBOL as a means for shifting a significant part of the burden of overall program specification from "technically"-oriented systems analysts/programmers to key corporate personnel whose competency lies more in the "business end of business" hasn't been realized by a long shot, despite widespread COBOL usage.

Now it seems to me inarguable that the grammars, symbol sets, syntactical protocols and so on of the synthetic languages we use to convey our intentions to a computer necessarily affect the faithfulness with which these intentions are represented by the programs we make. The impact on thought of closed systems of expression is the subject of long-standing debate; but the fact remains that we have no alternative to the use of such systems if we are not to fall completely silent.

Indeed, it could be said that all natural languages are themselves

closed, if only in terms of the numbers of things they allow us to describe: their finite alphabets ensure this. Even so, natural languages, however closed they may be when viewed in this way, are practically open and limitless, and it is that effective quality toward which we aspire in our synthetic languages. In our industry, we have acknowledged that it is properly our job to create the most natural-appearing synthetic analogies to the means we employ to express our *particularized* wishes. We have approached this challenge with some small measure of confidence, knowing that machines that need to understand "payroll" don't normally have to know what "love" means, too.

But COBOL's kinship to natural languages is almost solely cosmetic, limited to the sharing of a familiar vocabulary. Whatever benefits we may now claim COBOL to have provided through its alleged business-oriented nature, the participation of persons with lesser data processing capability than previously required hasn't occurred either.

That it has facilitated our usage of computer systems is, in my opinion, almost a moot argument, since COBOL promotes many phenomena which can only be characterized as undesirable. In addition to undoing certain educational achievements like proper spelling habits, it represents a ticket of admission to the systems analyst/programmer profession for many whose promise in applications development goes no deeper than the mere ability to create COBOL code. This is of as much value to increasing implementation productivity as the ability to read a foreign language aloud without knowing the meanings of the words might be.

Additionally, much damage was done by the misrepresentation of COBOL as a lingo so close to that commonly used by the world's businessmen that the need for program documentation would be satisfied by COBOL listings alone. This inflated claim quickly subsided upon our discovery that documentation development and maintenance, when essentially reduced to obtaining a clean final listing and keeping it clean, results in a *know-how* and *know-why* fade of the program's methods and purpose.

In some ways, COBOL's value is also misleadingly inflated by its status as an ANSI and FIP standard. Not that NBS or anyone else (except in highly specific individual cases) insists on mandatory COBOL usage in *all* applications, be they (so-called) commercial or otherwise. In fact, usage of the compiler is waning or will start to very soon — too late, alas, for many users to recover from the damage it may have done.

In December 1977, Dr. A. S. Philippakis, in an article entitled "A Popularity Contest for Languages," claimed that COBOL is "the most

widely and extensively used language for administrative data processing. . . ." Shockingly, he goes on to cite the Assembler-level language group as next in popularity. These and other general (but valuable) conclusions, supported by the mathematical bases underlying the methods employed to reach them, and further sustained by both direct and inferred qualifying commentary, constitute a sad historical footnote to the skill of our language designers.

With their intellectual snobbery and near-disdain towards the business community's needs, they seem unaware of the urgency of finding, between perfect structure and complete disorganization, reasonable compromises capable of yielding crucial, if interim, returns to users poised between COBOL and Assembler. Whilst we are treated to obscure, tiresome, theoretical lectures, thousands of writing pencils in the hands of thousands of COBOLoholics or Assembler stalwarts (who know job insurance when they see it), toil relentlessly on. We need far better tools than these Stone Age lingos, with their built-in suggestions that too much or too little is better than enough.

The Resurrection of RPG

As COBOL wanes, RPG usage is on the increase after many years of relative neglect. This is not a surprising development, especially in view of the potpourri of minicomputer products that emerged in the mid-1970s.

Small processors are incapable of handling a real, honest-to-goodness full-blown COBOL. Often their main mini-memories are either too small to remember enough at one time, or function so slowly, that recall for file passing is an out-to-lunch drag. I realize that this statement is likely to elicit cries of "foul" from our worthy mini-manufacturers, and protestations that they do, indeed, have "full ANSI COBOL" facilities in their literature. I am prepared to meet this probable response with Dr. Grace Hopper's COBOL test package criteria as my sword and shield.

Anyway, I think it would be a mistake to attribute to our very capable minicomputer industry the foresight needed to pull partly non-procedural RPG out of a hat, at the same time that language specialists were tiring of the proceduralized fruits of *their* most recent efforts. But the number of minicomputer companies rushing to provide some form of COBOL with their systems, as they churn out small versions of larger systems by the score, suggests that language is far down on their list of priorities.

Nonetheless, the provision of RPG with almost every mini, however accidental it may be to the rediscovery of RPG as a kind of "non-proceduralized" relief for people who want to program without attending to

too much science, has aided our quest for an improved user language interface. It will, however, be neither in the rigid artifice of COBOL, nor in the purely nonprocedural domain of RPG's possibilities that the answer to our language problem lies, but rather in some quasi-natural (that is, natural to someone) mix.

For the moment, we will limit our further attentions to RPG in order to make our point that procedural languages like COBOL, FORTRAN, etc. have missed the mark as desirable, long-term lingos. RPG experienced a quick rise in popularity in the IBM 1401/HIS 200-dominated marketplaces of the early '60s and a subsequent plunge to obscurity in the race to larger, faster machines that characterized the middle of that same decade. As larger machines appeared, we presumably were expected to be more orderly in specifying our needs by using more procedurally oriented development tools. Thus, we were "had" by our feelings of self-importance, since these grew proportionately with our exposure to the higher (and even more costly) forms this technology assumed. If ever there were a time to demand that computer systems shape up to be more responsive to man, this was it. We, naively, could not accept the implications of this sequence of events: could machines really think?

So, while most think of RPG's reappearance as having occurred by fluke, in that it was resurrected as a development processor because of the ease with which it could be installed on small machines unable to support a truly usable COBOL facility, the truth is profounder than that. RPG has proved to have more in common with our business-oriented ways of describing things than previously thought. The language feature underlying this commonality is the selfsame one cited as a reason for having discarded RPG in the first place: that is, its capacity to allow users to create programs in a nonprocedural way. By focusing the users' attentions on familiar-looking end result forms, pictures, etc., RPG has been found to satisfy the desire of those users to increase their participation in development more fully. But earlier on, RPG was criticized because persons writing RPG programs gave the appearance of preparing their programs in disorganized ways.

Thus, we note that those who invent modern computer systems hardware technology, with the intention (among others) of relieving us of highly demanding, people-consuming work, have achieved far better results in addressing the "highly demanding" area than in that related to "people consuming."

Incidentally, this led to furtherance of the 1960s myth that the computer system industry, *bottom line,* created more jobs than it lost. In

fact, an improperly conceived theory of how people could best transform heretofore manual white collar processes into automatic ones heavily contributed to this appearance.

Anyway, this orientation in systems development was furthered by our recognition of the fact that coding is but a small part of the applications creation process. The creation of functional specifications, systems analysis, testing and other non-language-related activities constitute over 80% of the task. But mere percentage computations are misleading here — it is *only* through our usage of today's language processors that we can cross the abyss between man and machine.

Now we can see why there is reinvigorated concern with language in computer technology. For it constitutes a more important communications channel than, say, the physical I/O devices between hardware components do: it links man to machines.

Language and the Man/Machine Interface

Speaking of software concepts as subjects of popularity cycles: one which is cycling upward again provides hope, through new insights, for a solution to many of today's software problems. It embraces the theory that we may reduce (essentially) all man/machine interface problems to those relating to the languages we make and use to tell machines *what* to do. And once this is proven, gone will be the necessity of caring about operating systems, hardware, and non-natural means of "specifying" and "encoding" applications (except for the software implementor, of course). Further to our language thesis, I believe that this can lead to an escape from today's professional challenge that we act like mouldable robots capable of learning to "fly anything." The falsity of this challenge must be recognized, and the mindset from which it arises discarded.

In order to do this, we must conceive systems which can respond to as natural a form of *our* language as possible. Indeed, I view the ideal future system to be language-driven (not that it couldn't be viewed that way before), to the extent that the task of specifying *how* will take a back seat to that of specifying *what.*

Insofar as the user is concerned, systems newspeak lingos will involve essentially *open* repertoires which *anticipate* the kinds of things he particularly wants. Furthermore, these lingos, however closed they may be, will appear to the user to be unbounded and open to enrichment. At another level, they will include powerful commands which mix both textual and geometrical data to trigger the applications creation processes.

Whatever forms our future languages ultimately assume, it is clear that our need to be concerned with a special operating systems lingo will effectively disappear. Except for those concerned with providing pure software and other technology-related services (e.g., network and database administrative personnel), the overwhelming majority of applications implementors will properly find their challenge where it ought to be: at the applications specification level.

To achieve the language desiderata outlined above, the study of language development techniques which enable us to tell machines what to do, and to receive reasonable cooperation from them in carrying out our wishes, is mandatory. Through this approach, great refinements of the man/machine interface will occur and even greater breakthroughs in programmer productivity will be achieved.

Man has employed algorithmic notations since Euclid's time. Many of these notations resemble programming languages in that they are intended to allow algorithms to be specified so unambiguously that they can be studied as mathematical objects. Other notations have far less need for formality.

When the purpose of an algorithmic language is to instruct a person (as opposed to a machine), and when it is likely that every specification of a function to be performed will be accompanied by its written description (in natural language) plus a picture of the end product, there is little need for mathematical precision: the person's interpretive powers are assumed. For example, if in executing knitting instructions (see Figure 10-1), an error is noted (either in the algorithm itself or in the language used to express it), an experienced craftsperson can make a simple adjustment and still achieve the desired result.

The luxuries of natural language — anaphora, redundancy, ellipsis, hyperbole, contextual interpretations of words — that can be indulged when writing knitting instructions are not available in current programming languages: but they may well be in the future. In "What the Nature of Natural Language Tells Us About How To Make Natural-Language-Like Programming Languages More Natural,"* the author, J. R. Hobbs of City College, CUNY, concludes that it is "reasonable to aim for programming languages that have the flexibility and richness of natural language and for programs that have the texture and coherence of a natural language paragraph."

---

*A paper appearing in "The Proceedings of the Symposium on Artificial and Programming Languages" as published in *SIGPLAN Notices*, Vol. 12, No. 8, August, 1977.

Within a short span of time, we have witnessed the development of sophisticated, high-level procedural languages, and, over the next decade, COBOL, FORTRAN and PL/1 (those old soldiers) can be expected to fade slowly from the data processing scene. A variety of powerful, problem-oriented, basically nonprocedural languages can be expected to take their place, (although the use of nonprocedural languages in pure systems software development is likely to be limited). Unless political forces keep it alive artificially, COBOL will be the first to go, even though it is the language most widely used today as an applications development tool. I suspect that programs originally expressed in it will be run long after their original source code has disappeared, with hundreds of millions of investment dollars poured into current applications.

FORTRAN and FORTRAN-like processors will also fade: in doing so however, they will spawn evolutionary sets of less ambiguous, more powerful, processors.

Whilst we await the computer newspeak of the 1980s, various other processors will be attempted. Some will follow in COBOL's footsteps; others will live periodic lives, to die or to give rise to others that die in their turn; others will blossom. The stimuli for birth, death and resurrection of higher level languages will be influenced by many factors, including hardware improvements, availability of R&D resources, and our own educational envelopes.

Whatever speed this evolution achieves, it is clear that we are headed toward more nonprocedural means of expressing our applications needs. Once this transition has been made, language designers will turn to more advanced techniques of artificial intelligence in order to achieve a truly natural method of expressing computer programs. Of course, such expression is but part of the problem of creating applications in economically and technically viable ways.

Having stated this, we may go on to explore these means, with the reader comforted that I do not consider them the only bottleneck in our software world; just one, major one.

The Progression From Procedural To Nonprocedural Languages

A programming language is a vehicle for specifying abstract mechanisms capable of processing data on an automatic computer. That programs must eventually be run on actual machines, quite naturally, has had more influence on computer language design than any other factor, including the need for precision.

The earliest programs were written directly in machine code. Symbolic assemblers were introduced when it was found — to nearly everyone's surprise — that machine code is hardly an effective means for specifying algorithms. Macro facilities were developed in order to reduce the volume of coding (without taking away the ability to use the most primitive commands in the hardware repertoire). The Autocoders that followed were one step above macro assemblers, offering algebraic notations approaching that of FORTRAN. With the introduction of FORTRAN, ALGOL and COBOL, the illusion of independence of algorithm specification from computer hardware was achieved. One could write a computer program without knowing much about the machine on which it would run.

But most programming languages in common use (RPG excepted) still conform to the conventional Von Neumann machine organization. They are intended to support primarily procedural programming; they provide for only a few data types (for FORTRAN: integers, real and complex numbers and dimensioned arrays); and they tend to encourage the use of constructs (such as the "go to") which diminish the clarity and manageability of the source code. Procedural languages give the user a sense of control over performance that would be absent in a language like RPG. Programmers are thus disposed to search for the most efficient — not the most logical — solutions, and they are encouraged in this pursuit by managers who worry only about machine performance. The supportive documentation that is issued for systems software usage is not very helpful either. A recent PL/1 text, for example, cautions the reader to avoid procedure calls whenever possible because they incur so much processor overhead. Who knows what sort of programming results from advice like this?

In Chapter 2 it was noted that basic software costs are steadily outpacing those for hardware — as much as 80% of the total hardware/software budget over the next ten years may be spent on software. Our current programming languages account for much of this imbalance. The waste that they cause is traceable not only to the awkward means they provide for the expression of our requirements today, but also to the means by which yesterday's code must now be maintained and operated.

It is only recently that the data processing industry as a whole seems to have recognized that the old standbys, FORTRAN and COBOL, have faults which go far beyond mere intellectual objection. There are damning limits in each which ensure our long-term dissatisfaction with the paths they provide us in our search to create *natural* man/machine interfaces.

The crusade for structured programming and the proliferation of preprocessors designed to improve programmer productivity attest to the

growing unrest with regard to these languages. Structured programming is a small but positive evolutionary step, and it can be expected to boost productivity to some degree. But it is an unfortunate reality that the structured programming discipline, however valuable it is, serves more to separate the professional from the amateur than to upgrade the amateur. However suffused in intellectualism it may be, it gives us a view to where things are going as we evolve our means of expressing applications. It is through the final incorporation of structuring facilities (e.g., system software algorithms) *into the CPU* that the amateur will be raised to professional status. As we noted earlier, nonprocedural processors place far fewer demands on the user, allowing him to express applications requirements more in terms reflecting the *end* to be achieved rather than the *means* for their achievement.

As for current "programming aids" packages intended to upgrade user abilities, most are really too trivial to merit serious consideration. The suppliers of COBOL preprocessors, in particular, have confused economy of expression with language improvement. The problems of today's programming languages are fundamental, and abbreviated notations are not likely to contribute much toward their solution. While helpful within certain limits, vocabulary changes alone do little to solve the basic issue of *how* to organize the applications procedural environments.

In an attempt to provide facilities approaching those we desire in nonprocedural processors, FORTRAN and COBOL are being hastily patched up. PL/1 has never really gained acceptance (perhaps because the low quality of early PL/1 compilers made lasting enemies for it, but clearly because of its excessively intellectual orientation). ALGOL, still popular in Europe, seems only to interest the academic mind in the U.S., and Assembly programming is clearly on the way out. APL, while enjoying a recent resurgence in usage, is more naturally suited to prototype or one-shot applications, where performance is not an issue at all—its interpretive code consumes too much time. The appearance of special microcoded "APL-Assist" engines help out, but not enough.

The only commercial programming language that is gaining in favor is the reemerging RPG. RPG is limited to a certain broad class of business applications; when applicable, it can be extremely effective.

M. Pauken, writing in the *Infotech State of the Art Report,* "Commercial Language Systems" is glowing in its praise. "Programming time can be reduced by approximately 60% from that required with a procedural language like COBOL. . . .Experienced RPG II programmers usually only take a few minutes to determine what changes to a program are necessary to meet a user requirement and then only a few more minutes to make the change. Our experience with RPG II has also

shown that an experienced programmer can write a basic RPG II program with no errors; the program will compile and execute the first time without error. Productivity is without a doubt the greatest strength of RPG II."

RPG II is said to be a problem-oriented language. The term "problem-oriented" is roughly synonymous with any of the following:

very high level	problem statement
nonprocedural	problem definition
less procedural	problem description
goal oriented	systems analysis
pattern directed	specification
declarative	result specification
functional	task description
relational	

Jean Sammet and Burt Leavenworth of IBM who compiled the above list say that each of these terms is intended to "convey an intuitive notion of languages which is in some sense 'higher' than FORTRAN, COBOL, PL/1, etc."* The important distinction for our purposes is that RPG departs from the Von Neumann digital computer organization which dictates that programs must be made up of a series of steps.

In order to write an RPG program, it is necessary to fill out five forms: the control and file description, extension and line counter, input, calculation and output specifications. Except for the calculation specifications, there is no requirement that an order of execution be given.

There is a definite relationship between the degree to which a language is procedural and the productivity of the programmers that employ it. Essentially nonprocedural RPG is widely considered to be an "easy" language, while assembly languages which are totally procedural are generally agreed to be "difficult." FORTRAN, COBOL and PL/1 are in between, being less procedural than assembly languages but more procedural than RPG. Computer scientists are now persuaded that progress in the field of language design can be traced by the movement from sequenced modes of expression to declarative modes of expression. They thus confirm the envelope of indecision within which today's languages were born.

It is true that programming languages have become less and less procedural. The FORTRAN expression

X = A+B*C+D

*From an article appearing in *SIGPLAN Notices,* Vol. 9, No. 4, April, 1974.

taken alone is not procedural. Yet this single unit of code requires many lines of procedure when it is expressed at assembly level. In Manchester Autocode (Brooker, 1956), one line of code was not sufficient to code the expression, but it was still possible to state the logic more compactly than at the assembly level:

```
X1 = B*C
X2 = A+X1
X  = X2+D
```

We see a definite progression from procedural coding to nonprocedural coding in the above examples and it is not difficult to show how PL/1, a more modern language than either FORTRAN or COBOL, represents a continuation of this progression. The PL/1 programmer is able, for example, to avoid coding what would have to be a loop in FORTRAN by employing operators that apply to entire arrays or data structures. There are similar facilities in APL, another relatively new language.

Databases and Data Representation

The debate between proponents of the CODASYL Database Task Group (DBTG) database management proposal and the advocates of relational database systems over what constitutes the most reasonable model for representing data of the sort found in business files has been in progress for several years. Fundamental to this debate is the question of whether or not the Data Manipulation Language (DML) should become less procedural. The DML associated with the DBTG model of data representation is essentially procedural, while relational database access is achieved by specifying what is wanted instead of how to find it.

In "An Analysis of the April 1971 Data Base Task Group Report, 1971 SIGFIDET Workshop Proceedings," Robert Engles of IBM argues the case against procedural data manipulation convincingly when he suggests that it should be possible to write the single statement

```
STORE DEPT-REC.
```

instead of the procedure

```
IF DEPT-REC< 3000 MOVE "AREA-A" TO DEPT-REC-AREA-ID
  ELSE MOVE "AREA-B" TO DEPT-REC-AREA-ID.
STORE DEPT-REC.
MOVE SUPERIOR-DEPT-NO DEPT-NO IN DEPT-REC.
FIND DEPT-REC SUPPRESS UP-SET CURRENCY UPDATES.
MOVE DEPT-REC-AREA-ID TO REL-REC-AREA-ID.
STORE REL-REC.
```

He points out that as "department records are uniquely identified by department number and each department record contains the number

of its superior department, the relationships among departments are completely and precisely given by the values of SUPERIOR-DEPT-NO. If the structural significance of this data-item were known to the database management system (through the Data Definition Language, or DDL), then it could perform the above procedure."

Engles' suggestion makes sense because it would reduce the volume of writing for the coder (though not for the database administrator), make the program more readable and therefore less of a maintenance headache, and permit the database to be restructured without impacting program logic. One of the greatest advantages of the database management concept is that in defining the database via a nonprocedural language (DDL), the database administrator specifies one time what would otherwise have to be coded for every program written that addresses the database. The more powerful data description languages become, the more it will be possible to avoid the repetition of "hard" procedural programming tasks.

The appearance of database management systems based on the CODASYL DBTG Report typifies the trend from procedural to nonprocedural languages. Here we see, for the first time, a clear distinction between the specification of data (which is envisioned as the responsibility of a centralized group or single individual) and the manipulation of data (which remains the responsibility of the programmer). When the programmer issues, for example, a "FIND" command (see the previous example), he activates a procedure that is determined by the content of the data declaration prepared earlier by the database administrator. The DDL employed by the administrator is nonprocedural; each statement in DDL relieves programmers of the burden of creating the procedural code found in other contexts.

The DBTG Report, despite its defects (so eloquently exposed by IBM), represents a significant advance in software organization. This advance is yet another example of the steady progression from procedural to nonprocedural modes of expression in computer programming languages. The fact that the report has most often been criticized because of its procedural DML indicates that the forces that have brought about the trend toward nonprocedural languages are not lessening.

Financial pressure and the relaxation of processing cost (attributable to the certain availability of high-speed processors and essentially infinite memories) are likely to combine to accelerate the trend. Software developers will be encouraged to come up with solutions to the problem of low productivity in the applications development and maintenance process. They will respond by designing very high level languages tailored to the needs of particular classes (e.g., industrial, functional,

etc.) of users. Dependence on unwieldy general purpose languages will diminish, there will no longer be a pressing need for super-fast compilers that generate highly optimized code, and breadboarding (experimenting with special purpose translators and compilers built quickly from standard components) will become commonplace.

What sort of languages will emerge from this activity? There will be three major types. The first type, to be employed by systems level programmers, can be expected to follow natural evolutionary lines for current procedural processors. The second type will be patterned on the Data Definition Language of the DBTG Report, languages for centralized administration of the database, languages for control of the communications network, for auditing, etc. The third type, geared to the needs of commercial data processing people, will resemble application generators driven by questionnaires and incorporating automatic prompting along with powerful diagnostics. "Natural English" declarative languages having English, natural language keywords are already being developed in universities.

Gaining the capability to trigger the creation of applications functions through the processing of keywords surrounded by redundant, ambiguous or otherwise extraneous words is our goal. *And the computer system of the 1990s may indeed be used to mediate between our varying images of the same application, to promote a kind of consensus as to what it "truly" is.* This discriminatory power brings nearer the day that artificial intelligence may leave the realm of the pure scientist to be at the immediate disposal of the businessman.

New Innovations

At MIT, questionnaire-based programming is receiving serious attention from the Automatic Programming Group headed by W. Martin. According to Leavenworth and Sammet in "An Overview of Nonprocedural Languages," "a significant part of their approach is the use of a questionnaire which is carefully designed for a particular specialized application area. The user will answer the questions, and also communicate more information about the details of his problem in 'natural English'."

At the University of Michigan, formal language theory plays a more important role. The ISDOS project, headed by D. Teichroew and A. Merten, has produced a Problem Statement Language (PSL) in which the user states his requirements in top-down style, progressively refining his declaration until a suitable level of detail is reached.

Many other efforts are underway that are equally revolutionary in scope. It has taken years for scholars to take an interest in the problems

associated with commercial data processing, but now they are likely to increase their efforts. For one thing, there will be more and more research funding available for this kind of work. Current statistics which show the alarming lag of software development technology in comparison with hardware will cause this, if nothing else does.

For another, software technology has advanced to the point where truly interesting cross-disciplinary efforts are feasible. All of that research in formal language theory during the 1960s and early 1970s is paying off now; the construction of language processors is no longer the formidable task that it once was and implementors are looking for ways to apply their new techniques.

It has always been easier to create programming languages for mathematicians, engineers and computer scientists than it has been to create commercial data processing languages. The following list (by no means exhaustive) is indicative of how responsive language designers have been to the needs of their fellow scientists:*

1. Arrays and their various combinations and transformations.
   Example: APL.
2. Lists, and their recursive evaluation.
   Example: LISP.
3. Sets, maps, and their various combinations.
   Example: SETL.
4. Strings, patterns, and the matching operations which relate them.
   Example: SNOBOL.
5. Electric circuits and their interconnections.
   Example: LINL, and other similar electrical engineering languages.
6. General classes of objects interacting through discrete events.
   Examples: SIMSCRIPT, SIMULA, GPSS, and various other discrete simulation languages.
7. Algebraic formulae, and their various manipulations.
   Examples: FORMULA ALGOL, MATHLAB, and other languages for algebraic manipulations.

---

*"Semantic Definition Methods and the Evolution of Programming Languages," Jacob T. Schwarts, *Formal Semantics of Programming Languages,* Randall Rustin, Editor. Prentice Hall, 1972.

8. Elastic structures formed out of thin members, and their interconnections.
   Examples: STRESS, COGO, and various other mechanical engineering languages.

9. Generative grammars, syntactic types and their subgrammars and strings.
   Examples: Various language-describing metalanguages.

10. Plane or three dimensional figures, and their geometric combinations and relations.
    Examples: Various languages for figure production and analysis.

Many areas of mathematics and engineering happen to be well suited to treatment on a digital computer because they have been formalized and reformalized and refined over many years. The activities of government agencies and corporations — even in narrowly defined areas such as accounts receivable and payroll processing — have never been rigorously specified (attempts to create generalized payroll and accounts receivable packages have been among the most frustrating and amateurish endeavors of the software industry*).

In addressing the problems of industry, software specialists have tried to impose incomplete and often incorrect models of an enterprise because their knowledge of it is frequently so superficial. The interdisciplinary teams that are producing the most interesting research in programming languages today are approaching their work from a much more knowledgeable and realistic point of view than did their predecessors. The gap between the designers of software and the enterprises that their products will eventually describe has narrowed perceptibly in this decade.

Manufacturers, spurred by the success of RPG and RPG-based applications generators, are intensely researching the possibilities of essentially declarative special purpose languages. Since it is possible to create useful application generators quickly and inexpensively using macro processing techniques, many software houses are planning to concentrate their own software development efforts in this potentially lucrative area.

---

*Many problems in this area relate to our inability to understand and capture continuously changing phenomena in concrete forms. Our acceptance of things as essentially continuous principles which achieve "completion" as temporal phenomena comes too late. I think that a shift in our preoccupation with "managing change" to one of learning to manage "prototype scaffolds" within which change would be accepted as a natural phenomenon would solve many problems in this area.

By the end of this decade a substantial number of new application-oriented, nonprocedural languages will become available. During the 1980s there will be a proliferation of such languages characterized by progressive growth in sophistication. It is likely that early offerings of this type will not, in the strictest sense, be languages at all, but merely specialized notations for invoking a set of standard services. Some of these are in their earliest (or purely experimental) forms today; others found in and about information retrieval are quite advanced. With basic software providing the procedural glue to sets of nonprocedural declarations, a major bottleneck in program development will be overcome. With structured and managed databases, powerful mixed relational/hierarchical/network techniques, massive memories (likely by 1985), and hardware (including communications) operating at sub-nanosecond speeds, programming power will be available to users with any level of sophistication (in computer technology, that is).

Implementation Languages and the Case for PASCAL

Although implementors' languages have been in use for some time (Burroughs pioneered the use of higher level languages for software development in the commercial arena), programmers have resisted using them and have insisted that they be sufficiently low-level to permit the most basic facilities of hardware to be invoked. The propensity (if not need) of systems analyst/programmers to think more in terms of how systems programs can best be optimized has not helped matters at all. Language designers have been in a defensive position because they are constantly challenged to prove that their systems will generate code competitive with what might be created through use of Assembly language. It is not surprising, therefore, to see the oddities of particular machines faithfully reflected in a higher level implementation language.

There are signs that the traditional barriers to progress in this area are eroding. Because there is almost universal agreement in the software community that something like structured programming (the term has never really been defined) is necessary, implementation languages are now being taken more seriously. The work of men like Edsger Dijkstra, C.A.R. Hoare, Per Brinch Hansen and Niklaus Wirth, and the entry into the field of a new generation of university graduates free from the old prejudices have helped bring this about.

As an example of one of the latest implementors' languages, many computerniks are currently touting PASCAL. The fact that its name is (a) not an acronym and (b) French, is a refreshing sign of maturation in the software-producing community on the one hand, and of newly emerging class consciousness on the other. But while no one claims PASCAL

to be the ultimate developer's tool, it seems to be emerging as one of the best available.

PASCAL, a successor to ALGOL 60, is familiar to most recent graduates in computer science. It was planned as a vehicle for teaching programming (and good programming habits) and it caught on like wildfire in the universities. In 1972, when Dr. Niklaus Wirth, the developer of PASCAL, was at Stanford, he delivered a number of talks on the language throughout the United States. Before Dr. Wirth returned to Switzerland, the University of Colorado agreed to act as a distribution point for PASCAL in this country.

The early growth of the language can be illustrated by the number of CDC sites installing PASCAL over a three-year period (the first PASCAL compiler was written for the CDC 6000 series).

- 1972 14
- 1973 22
- 1974 46

The PASCAL compiler has been rewritten so that it can be transported with ease to new hardware. There are versions for the CDC 3300 and 3600, the SIGMA 6 and 7, the PDP 10, PDP 11/20 and 11/45, the IBM 360, 370, and 30XX, the B4700 and B6700, the Telefunken TR440, the Phillips P1400, the ICL 1900, the CII Iris 80, the DEC System 10, the Univac 1108, the Data General 840, and the Honeywell H6000 series.

PASCAL is not supported as a standard product by most computer manufacturers. However, its consideration as such is gaining momentum, with some manufacturers providing informal support to user-originated versions. CDC has implemented a cross-compiler which permits programs to be written for their front-end processor and compiled on the 6600, but this version is for internal CDC use. The language has been entirely supported by Dr. Wirth and five or six associates, who operate an informal agency that maintains the compiler and issues occasional newsletters. At the moment, PASCAL is an "underground" implementors' phenomenon, a kind of *Volkssprache.*

Why has PASCAL achieved such underground success? There was no comparable rush to use PL/1 when the well-publicized MULTICS project decided to use that language. A number of reasons were given by G. Richmond writing in the February, 1976 issue of *SIGPLAN Notices:* "PASCAL is a good language for teaching programming and the development of software. . .[it] is a much more readable and versatile language than ALGOL or FORTRAN. It provides the control and data structures that are desirable for structured programming. . .Its portability

to other computers gives it the potential to be as widely used a language as FORTRAN."

The concept of *data type* is central to the success of PASCAL. The language offers four primitive data types: integer, boolean, character, and real, plus facilities for constructing new types. For example, it is possible to specify a "shape" data type or a "color" data type if shapes and colors can be conveniently used to solve a given problem.

Once a data type has been established, one can construct procedures that operate on objects of that type. Programs can thus be written in terms of *meaningful* abstractions rather than in terms of abstractions that are imposed by a language designer.

PASCAL is more interesting than other influential, new development languages such as ALGOL 68, because it is apparent that *it was designed for software engineering purposes.* This is revealed clearly in Wirth's book, *Algorithms + Data Structures = Programs*, in which a series of problems are posed and solved in PASCAL by the technique known as "stepwise refinement." The book is recommended reading to those who doubt that anything other than an "official" language system should be used to attack large programming tasks.

Concurrent PASCAL, an extension of PASCAL, was developed by Per Brinch Hansen of Cal Tech and later used to code the SOLO operating system. Hansen is certainly pleased with it. "I wrote the operating system and its utility programs in three months. In machine language this would have required 20-30 man-years and nobody would have been able to understand the system fully. . . . From a manufacturer's point of view it is now realistic and attractive to replace a huge and ineffective 'general purpose' operating system with a range of small, efficient systems for special purposes."

In February, 1976 a survey of the people on the PASCAL mailing list was conducted. The results showed that the language is used nearly as heavily for software development as it is for educational purposes. Perhaps the most interesting result of the survey was an unexpected one: several respondents, "mainly those associated with the computer industry," asked that their replies be kept confidential ("PASCAL Newsletter," February, 1976 *SIGPLAN Notices*). PASCAL apparently has secret admirers outside of the academic world.

Administrative Languages

Figure 10-2 shows the conceptual architecture for a database management system ("1971 SIGFIDET Workshop" publication). The schema

is a description of the totality of the database in some Data Definition Language (DDL); the sub-schema description must be in some sense compatible with the subset of the schema to which it relates.

One of the major strengths of the database concept is that it places the control over the centralized database in the hands of a single individual or group, called the database administrator (DBA). The administrative language (DDL) employed by the DBA is expected to receive considerable attention in the near future.

First of all, the concept of an administrative language can be applied profitably in many areas. For example, a programming standards administrator could employ a special language to define programming practices within his organization; he would prepare a programming standards declaration and submit it for compilation; the compiler would then produce a standards document to be issued to all programming personnel, plus a computer program that would be used to scan source code to determine whether or not the standard was being followed. If the program were to detect a standard violation, it would produce a report or annotate the programmer's listing.

Secondly, new developments in database technology will have to be reflected in the Data Definition Language. Figure 10-2 depicts a system in which the database is physically, as well as conceptually, centralized. In order for distributed database technology to become a manageable reality, there must be some way for the database administrator to define how the data is spread across a network of processors connected by communication lines. The creation of software for a system of physically distinct cooperating database managers would be incomplete without adequate language processor support.

Finally, while the centralization of database administration solves many problems for an enterprise, it creates others. A fundamental principal in business organizations has always been separation of duties. Placing all of the responsibility for what amounts to the most valuable corporate asset in the hands of a single individual or group invites fraud, among other things.

Data Definition Languages may have to adapt to reestablish the traditional, internal system controls without sacrificing the advantages of centralization; they may also have to adapt to satisfy the requirements of external auditors who do not like the idea of accessing data through the system rather than directly by means of their own software packages. In the current formulation of database management systems the auditor will simply be a user of the system. Like any other user, he will be dependent in a very real way on the database administrator for access to the files. He can never be certain that he is seeing all of the data.

The solution to these problems is not apparent at the present time. It is clear, however, that whatever the solution is, the design of Data Definition Languages will have to adapt to accommodate it. It is also likely that entirely new administrative languages will be created to support external audits of the database.

Artificial Intelligence

In Spring 1977, David Levy, an international chess master, defeated Chess 4.5, a computer program running on the CDC Cyber 176, in a surprisingly hard-fought game which took over three-and-one-half hours. Analysts claim that as late as the seventeenth move, Chess 4.5 had the advantage, and that the outcome had remained in question until the computer blundered on the twenty-sixth move, (see Figure 10-3).

After that game was over, a second game was played in which the participants had only five seconds to make a move. This time, Chess 4.5 won in 38 moves.

The program was "tuned up" for its match with Levy by winning the Paul Masson Class B Championship in convincing fashion from its human opponents who, in order to qualify, had to be ranked Class A or Expert (a Class A player has a rating of 1800 to 1999; David Levy is rated 2325, and the current World Champion is rated 2690).

The application of heuristic theory to the development of chess program algorithms has resulted in a rather startling display of one kind of machine intelligence, suggesting the power which may ultimately be at our disposal when machines with such thinking capabilities are commonplace.

It is mind-boggling to consider the extent to which areas of activity once reserved solely to human beings are regularly being penetrated by advances in computer systems technology. The ability to "see," "speak," move about and so on, while far more restricted than that found in people, is nevertheless becoming increasingly "ordinary" in modern systems.

AT&T Long Lines' headquarters, Citibank's headquarters and many others now employ sniffing robots which, following a pre-sprayed course, routinely deliver the mail, beeping in annoyance at human beings who stray across their path at inopportune moments, and rewarding others with a cheery call to pick up their correspondence at assigned stations.

Of course, the activities of these mail robots are more hardware- than software-driven. While they exhibit a basic form of artificial intelligence, little more than the same genius in engineering already proven

in, say, elevator development was needed to bring them into being. What has been missing (among other things) from the products of our quest to deploy more comprehensive forms of artificial intelligence is the *synthesizing* capacities of our own brain. By this I mean the ability to assess and act upon a complex admixture of internal and external stimuli — perhaps even to decide that a given employee should not, at a given moment, receive a particular piece of mail!

The creation of software techniques which contribute to our ability to endow computers with intelligence has been underway from the moment computer technology itself began. However, it is only in the last ten years that really significant advances have been made.

The availability of more powerful devices incorporating mass memories, broadband, high-speed communications and an ever-increasing variety of I/O devices has created new opportunities in the exploration and expansion of artificial intelligence, and led to the resolution of heretofore insoluble problems.

And, while the chess-playing computer that learns not to repeat its own (or its opponents') mistakes seems to be of about as much practical use as chess itself in the conduct of our lives, it is what the *process* of computer chess-playing suggests that holds the greatest promise.

Chess 4.5 is a rather spectacular example of a computer program that mimics the higher mental processes of human beings. Whether we use the term "mimic," "simulate" or even "replicate" in describing what the computer system does seems irrelevant to me. I find no cause for either alarm or depression in the idea that machines may surpass men in doing *anything,* including thinking.

Indeed, I believe that such negative reactions to the potential of machines reveal too exalted a sense of respect, even of awe, for machines, and not nearly enough for the resiliency and richness of human beings!

With the emergence of artificial intelligence, we are confronted, more or less suddenly, with a reality that challenges many of our ideas concerning the very meaning of intelligence and the nature of man. But the scientist does not suggest that *our* relationships with machines need mimic, simulate or replicate those we enjoy with other human beings.

In fact, both the hardware architecture and basic software characteristics of today's systems truly require that we adjust to working in "non-natural" ways to create our applications. With more intelligence packed into our systems, such adjustments will, increasingly, be the

systems' problem, not ours.

You might say that CPU intelligence brings human freedom in its wake.

Furthermore, no scientist works to replace men with machines. Rather, he works to shift the burden of human *toil* onto the shoulders (if you will excuse an unfortunate anthropomorphism) of machines that do not tire. If we could learn to relinquish our fear of the machine, based as it is upon lack of confidence in man, we would recognize that such a shift can enhance the quality of our lives through the extension and rededication of our powers to other, more creative domains. In any event, I believe that the work of scientists involved in the development and refinement of artificial intelligence merely confirms the dual vision of man as both spiritual vessel and hapless prole: between these poles, we should have far less difficulty than some of us seem to have in acknowledging our need for machine-assisted or -lubricated improvement.

Pattern recognition, learning, problem-solving, theorem-proving and game-playing programs are being written with increasing frequency — and success — now that new, artificially intelligent languages and sufficiently powerful computers are becoming available.

Computer programs in the field of artificial intelligence normally consist of two parts; the pre-programmed part and the self-organizing part. In order to enhance the extremely poor performance of these programs, the tendency has been to overdevelop the pre-programmed part. This is an extremely labor-intensive approach that must be abandoned eventually. The increase in raw processor power that will characterize the next generation of computers should prove a stimulus to this.

While I do not think it likely that the office of corporate ontolgist or epistemologist will ever be created, I do believe that the study of the relationships between mind and object, the study of thought, ideas, imagery, perception and sensation, will influence the world of EDP greatly.

The potential influence on the character of programming languages is already apparent, in fact. Consider the efforts of John Ulrich of the University of New Mexico and Robert Moll of the University of Massachusetts, who are investigating program synthesis by analogy; the work of Zohar Manna of Stanford and Richard Waldinger of Stanford Research Institute, who have devised a deductive technique for automatically constructing recursive programs; the knowledge-based code generator (XGEN) written by Christopher W. Fraser of Yale University, and the PSI program model building tool under development at Stanford University.

The compiler of the future is more likely to resemble a "semantic learner" than a structured, unforgiving automaton, and I, for one, am delighted.

We noted at the start of this chapter that, contrary to the point of view generally expounded throughout this book (i.e., that change in computer technology is increasing in amplitude and frequency), we have not witnessed a burgeoning of innovation and growth in the software arena to compare with that typifying the hardware component of the EDP technology spectrum.

Recently announced products do incorporate startling advances in microcoded firmware, but mostly in the hardware control areas. Still, the current capability to design systems which apply chip technology (purely hardware) solutions to operating requirements previously addressed only by software, or to use such technology in conjunction with high-speed processors running microcode (firmware) represents a meaningful advance in software and applications development tasks.

The cleverly designed system that employs pure-software/firmware/pure-hardware mixes to enable us to enjoy a greater degree of creative freedom in our implementation and operation of software and applications is no longer unusual. Additionally, transformation of pure software into ingeniously engineered circuitry relieves both manufacturers and users of costly maintenance activities. But the positive impact of this technology is more directly related to hardware operation and pure software development than it is to the end-user's problems. In fact, the actual coding of microprograms is done through the use of techniques which date back to the 1950s. And while less work may be required of the operating systems development teams, the availability of mixed software/firmware/hardware facilities does little to change the *nature* of what they do.

Furthermore, as stated at the start of this chapter, it is certain that the operating systems we know today will disappear; that is, while their functions will remain, the means by which such functions are performed will be redefined and relocated. The extent to which present users need to concern themselves with the technology employed in today's wide variety of operating systems is perhaps the most significant measurement of the mammoth changes in computer technology that are still required. And while no one can doubt that with the availability of modern operating systems a great leap forward in our ability to create the conditions for symbiosis in applications systems was taken, it now seems clear that this leap was bought at too high a price.

Many people forget how difficult it was to produce a set of programs related by complex invocation sequences, data management and

other necessary adjuncts, let alone produce some sort of resource allocation/job accounting capability before today's OS, VS, MCP, GCOS, etc. environments saw the light of day. But along with the advent of these environments came new problems; learning an additional language (e.g., JCL) and, alas, the need for continuing education to remain up-to-date in the individual quirks of the system's control over its hardware/software/applications mixes.

In most systems, these idiosyncrasies give a convincing imitation of infinitude. When they *can* be compassed, they turn out to be of such significance to the user that he is obliged to keep up his guard by spending seemingly endless sums to insure that his facility maintains itself in a reasonably current mode.

In order to understand their systems, computer people must spend an inordinate amount of time attending conferences where they beseech manufacturers to explain the rationale for their operating systems architectures, learn about enchanting operating system foibles through the exchange of horror stories with their colleagues, and generally commiserate with one another over other systems exotica that they ignore only at their peril. This, it need hardly be pointed out, is time which might better be put to other uses.

With our shortage of skilled technical personnel increasing as computer technology proliferates, lack of operating systems know-how is inhibiting our ability to exploit our computer systems potential fully.

Clearly, the one way to put an end to this problem is to pack more intelligence into the CPU. Earlier, I noted that the availability of massive memories and truly high-speed processors will help make this possible by providing facilities that eliminate our need to be concerned with hardware efficiency (if nothing else). But the software we now write, maintain and use in connection with operating systems is only a crude model of what we can soon expect in this aspect of the man/machine interface area. There is no reason for the user to learn two different, non-natural protocols to express his application. For imbedded in the natural language facilities used to tell machines what to do will be the implicit license, granted without too many restrictions to the CPU, as to *how* to do it.

Our preoccupation in this chapter with the ways in which we must speak to a machine should now, we hope, take on additional meaning.

Future operating systems must be very much smarter in carrying out their jobs, thereby once and for all putting to rest our need to compromise our natural inclinations in order to accommodate the non-natural man/machine interface these operating systems currently impose.

The creation of artificial intelligence, our recognition that the most effective systems we can make must accommodate our many imperfections, and our acceptance of the virtue of achieving economy in the use of systems will go a long way toward shaping our software/applications development activity in the future.

Knitting Instructions "Program"

Using No. 11 needles cast on 130 sts.

Work in K.1, P.1, rib for 3 ½ in.

Next row Rib 9, (inc. in next st., rib 6) 16 times, inc. in next st.., rib to end (147 sts.)

Change to No. 9 needles.

*1st row P.5, * (K.4, P.3) twice, K.1, P.2, K.1, P.3 rep from * to last 16 sts., K.4, P.5.*

*2nd and every alt. row K.5, * (P.4, K.3) twice, P.1, K.2, P.1, K.3, rep from * to last 16 sts., P.4, K.3, P.4, K.5.*

*3rd row P.5, * (K.4, P.3) twice, C.3, P.3, rep from * to last 16 sts., K.4, P.3, K.4, P.5.*

5th row as 1st row.

*7th row P.5, * C.7, P.3, C.3, P.3, rep from * to last 16 sts., C.7, P.5.*

9th row as 1st row.

11th row as 3rd row.

12th row as 2nd row.

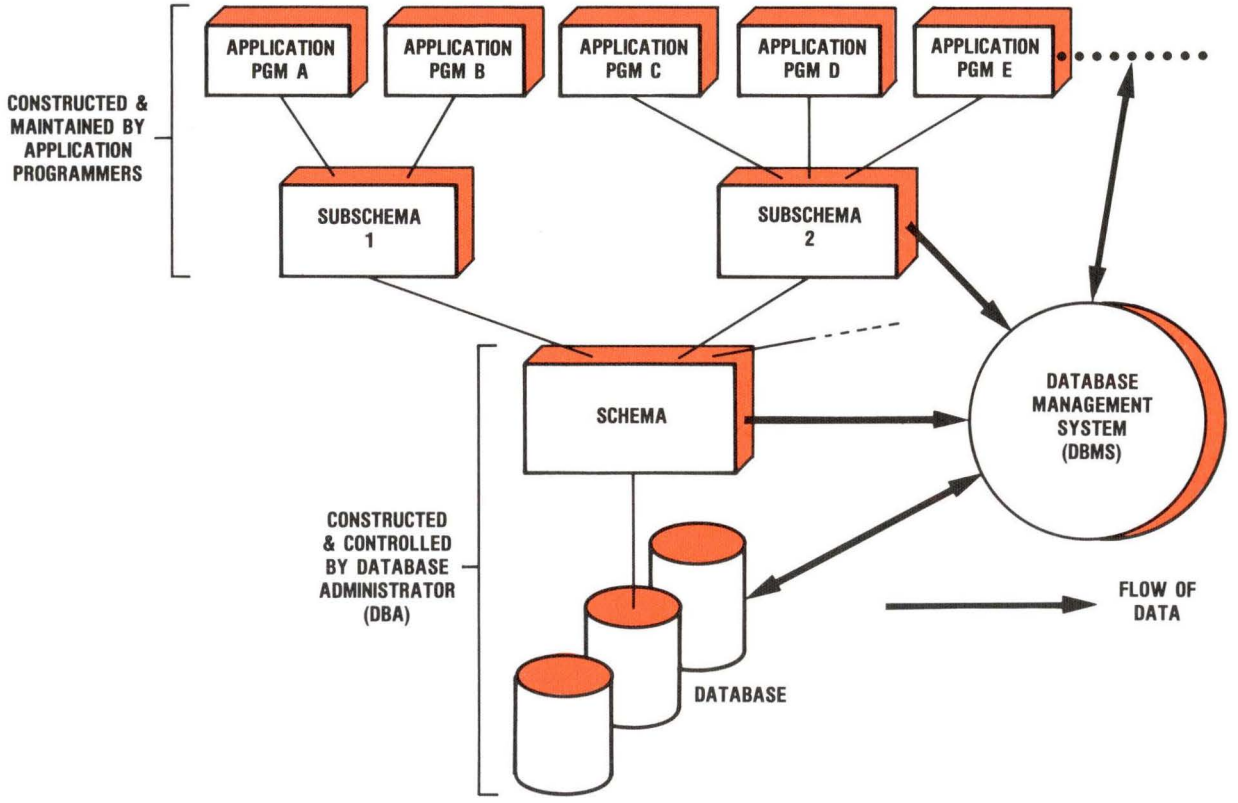
These 12 rows form the patt.

Continue in patt. until work measures . . .

SOURCE: A Comparative Study of Programming Languages by Brian Higman. American Elsevier, 1967.

Figure 10-1

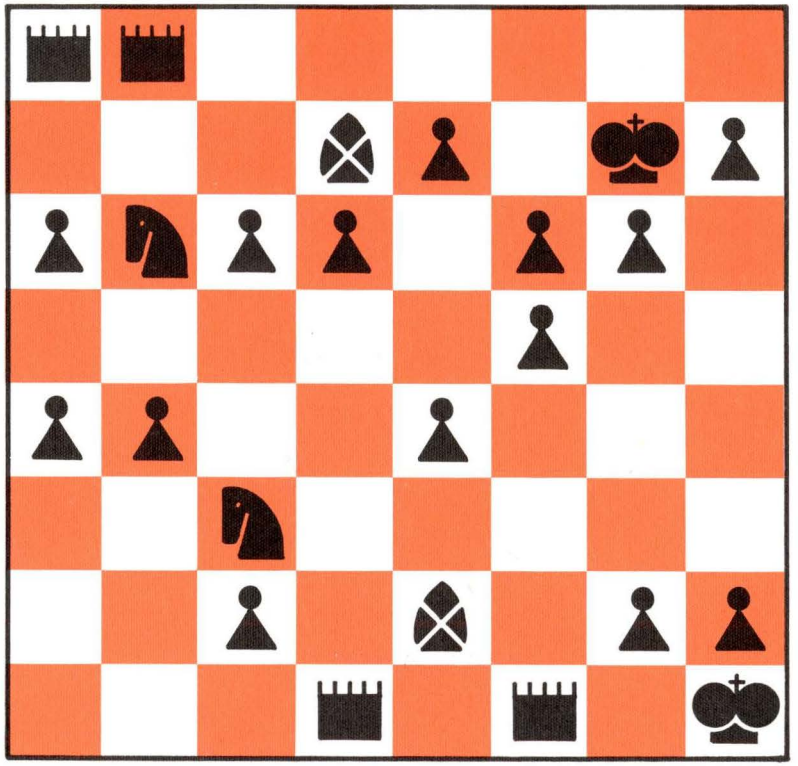
An Architecture for a Database System



SOURCE: "1971 SIGFIDET Workshop" publication, redrawn by Advanced Computer Techniques Corporation's Technology-Analysis Group (TAG).

Figure 10-2

Chess 4.5 (White) v. David Levy (Black)



1. P-K4	P-QB4		23. R-QN1	N-N3
2. N-KB3	P-Q3		24. R/B3-B1	R/B-QN1
3. P-Q4	PxP		25. R/N-Q1	P-B3!
4. NxP	N-KB3		26. P-QR4??	P-QR4!
5. N-QB3	P-KN3		27. P-N5	PxP/N4
6. P-B3	B-N2		28. PxP/N5	R-QB1
7. B-K3	O-O		29. R-Q3	R-B4
8. Q-Q2	N-B3		30. R-N3	R/R-QB1
9. B-QB4	P-QR3		31. R/1-B3	P-R5
10. NxN?	PxN		32. P-R4	P-R6
11. O-O	N-Q2		33. PxP	PxP
12. P-B4	N-N3		34. R-K3	B-K3
13. B-K2	B-K3		35. P-R5	P-N4
14. P-QN3	N-B1		36. N-Q5	P-R7
15. P-QR3!	Q-R4		37. R-QR3	BxN
16. P-QN4	Q-B2		38. PxB	RxP/B7
17. P-B5!	B-Q2		39. B-Q1	R-Q7
18. B-R6	Q-N3 ch!		40. K-R2	R-B8
19. K-R1	Q-Q5		41. B-N3	P-R8=Q
20. QxQ	BxQ		42. RxQ	RxR
21. R-B3	B-N2		43. Resigns	
22. BxB	KxB			

CHESS 4.5/WHITE POSITION AFTER 26. P-QR4??

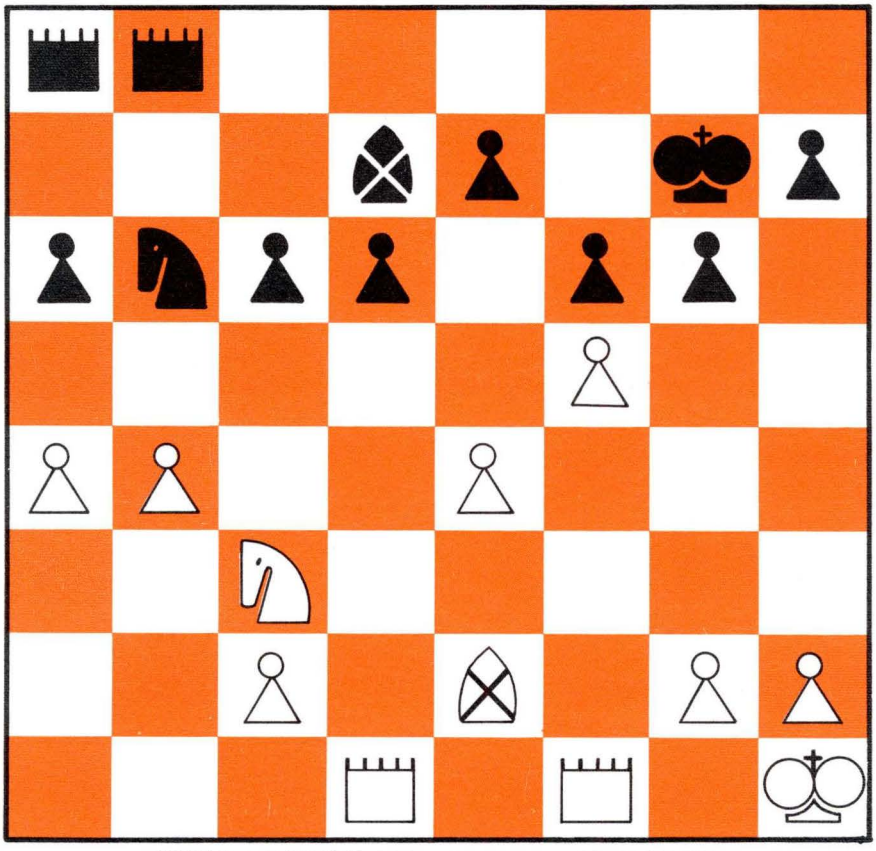
SOURCE: Illustration from "Science and the Citizen," Copyright© June 1977, by Scientific American, Inc., All rights reserved.

Figure 10-3

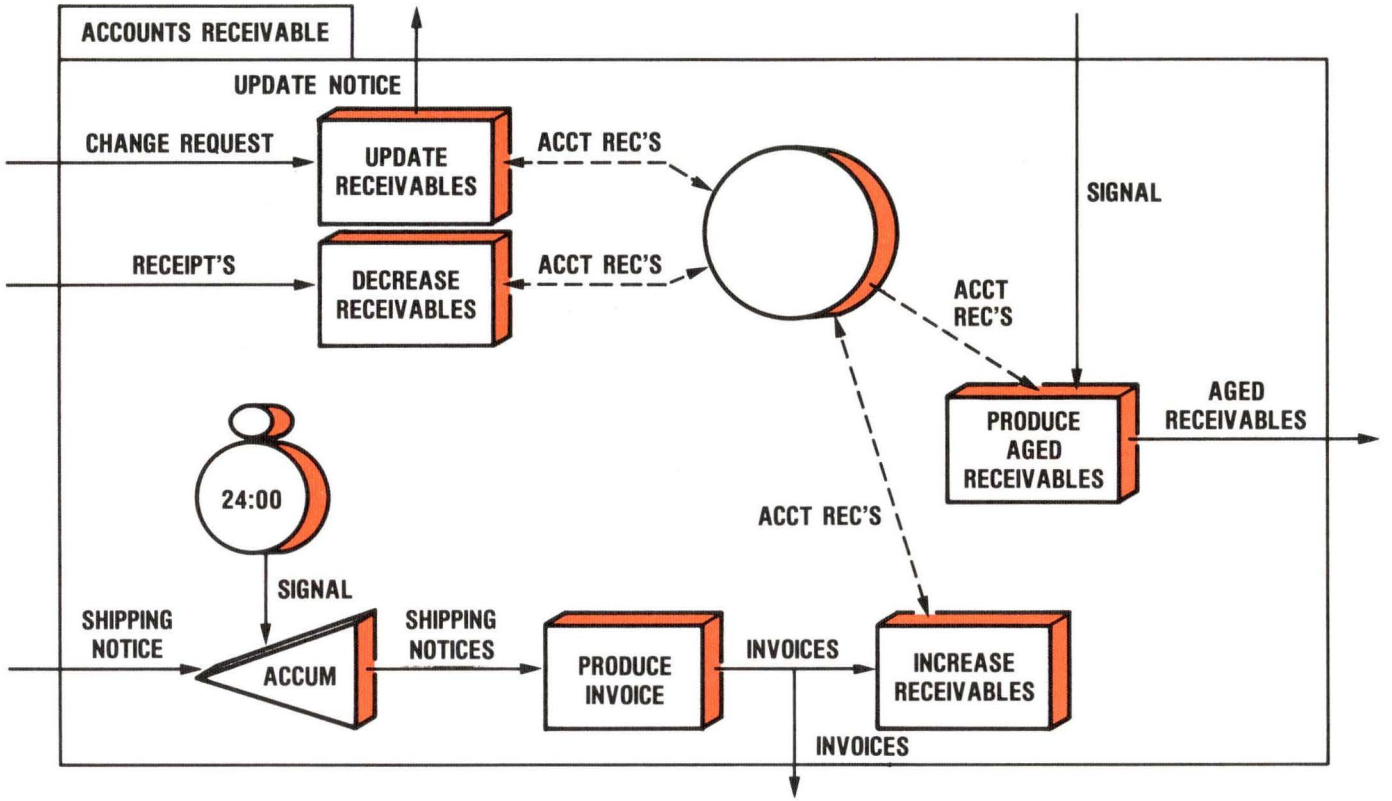
ERRATA

The disposition of the black and white chessmen in Figure 10-3, page 163, should be as shown in the diagram below.

Chess 4.5 (White) v. David Levy (Black)



CHESS 4.5/WHITE POSITION AFTER 26. P-QR4??



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 10-4

11. HARDWARE TECHNOLOGY TRENDS

We may well approach our crystal ball with the statistics reflecting past advances in computer technology in mind. They have been impressive, not to say incredible:

- Hand-held calculator costs have declined a hundredfold in 10 years.
- The cost per bit of random access memory (RAM) has declined by an average of 30% to 40% per year since 1970.
- Annual usage of logic gates (functions) or bits of memory has increased 2,000 times in 17 years (i.e., in 1960 approximately 500 million transistors were produced, while by 1977 an estimated one *trillion* equivalent functions or bits were manufactured and shipped).
- The number of components per integrated circuit (IC) has nearly doubled *annually* since 1960—from one component per circuit in 1960 to 256 bits or components per IC in 1967 to 16,384 components per IC in 1977.
- Disk drive capacity has jumped from five megabytes in the early '60s to over 500 megabytes today. Price per megabyte has also decreased nearly a hundredfold during this same period.

This very limited but highly representative sample of dynamic improvements could lead some observers to conclude that there is little room for further advances, and that the really significant developments must be behind us. This chapter will, I hope, demonstrate that such conclusions are false and, in terms of electronic logic and storage, "future shock" shall not truly be upon us until the mid-1980s (assuming that political, economic and psychological forces do not work to delay it

beyond its "natural term").

Nonetheless, today's technological environment, when viewed as a "happening" quite divorced from politics, psychology and economics, is indeed one of increasing transience. The seemingly continuous stream of product announcements, some coming from heretofore little-known sources, suggests an accelerating orchestral staccato.

New innovations, hoped for yesterday and ordered today to be delivered tomorrow, have a high probability of being called obsolete before installation. By obsolete, I do not mean useless. Rather, better alternatives will be made available. With increasing frequency, users and vendors are being subjected to the "had you waited a big longer" phenomenon in reaching and evaluating their decisions.

But the din of product announcements also suggests that the computer systems manufacturing organizations are "onto something big" in their quests for market position. Quite naturally, they must exploit their innovation continua to the fullest.

Users must reassess their ways of perceiving data processing product and services procurement policies, or they stand a good chance of seeing their DP budgets being inadvertently siphoned off for the good of technological progress with little to show for it. For the conspiratorially inclined, this may look like a concerted plan of the computer industry; nothing could be further from fact. If for no other reason, the industry isn't that organized.

Not unlike the dilemma facing the broker with a seat on the exchange who must assess when a stock will bottom or top out, the EDP manager is faced with buy and sell decisions of similar complexity. An increased interest in today's technological trends can go a long way towards resolving his difficulties.

But, in a sea of controversy over the ultimate ends towards which these trends are driving, remaining cool and conservative is mandatory. For the extrapolated end points of today's trends are absurd. These all seem to suggest that infinite capacity at zero cost will occur in the latter '80s. We know that's not going to happen.

While the user population may be experiencing instability in EDP planning, such instability caused by the increasing amplitude, frequency and impact of today's waves of change, the systems manufacturers are far more sure of what's happening and where it's all going. Their difficulties lie in accepting and reacting to change when it is adversely affecting their individual businesses, and in accepting and capitalizing upon such change when it is not.

The research departments of those who can afford research are steadily racing towards fulfillment of their goals in improving systems intelligence, reliability, accessibility and serviceability—all of this in delicate balance with economic viability. Some of the major areas commanding their attention follow.

Processor Performance/Price Trends Vis-a-Vis the IBM 30XX

General purpose (GP) processor performance can be measured in terms of "raw" MIPS ratings—add time or number of multiplications executed per second. Obviously such comparisons do not take I/O capability or software efficiency, not to speak of user environment scenarios, into account, and therefore should not alone comprise the yardstick used in the evaluation of "systems."

However, it is true that in evaluating computers which share similar architectures (as in the case of the IBM 370 series, for example), these comparisons have greater validity insofar as the establishment of relationships or processor performance ratios is concerned.

The graph in Figure 11-1 shows that "large-scale" general purpose processor performance has increased from about 20,000 instructions per second to nearly 5 MIPS (or a 250-fold improvement) from 1958 to 1978. During this 20-year period, note also that the *rate* of performance/price improvement has itself accelerated.

For example, 1953 to 1958 experienced a fourfold decrease in price per 100,000 executions/second; during the next six years (1958-64), this same performance level was available at 20% of 1958 prices; and over the next eight years (1964 to 1972), another tenfold decrease in price (or increase in price/performance, to put it another way) occurred.

From 1972 to the present (1977), however, there was only a two- to threefold improvement in price/performance (by IBM's 3033, 3032, 3031, etc.). Does this represent a possible levelling off of processor price/performance improvement for the future? This "mere" two- to threefold improvement in CPU price/performance (chez IBM, Amdahl, etc.) during the past five years seems to represent a temporary condition and tends to verify that the 3033 (and 3032 and 3031) are simply "interim"—if not end-game—products. (See our discussion on the 3033 in Chapter 8, "Short Life Span.")

In fact, the new IBM 3032 and 3031 both fit on the 3033 price/performance curve and were predictable. Other multiprocessor versions of the 30XX are also expected by the second quarter of 1978.

While circuit (ECL) packaging and heat challenges may slow down the rate of MIPS improvements for single processor configurations (until the System/80 is introduced in 1979-80), multiprocessors (two to four CPUs) should compensate; i.e., each additional CPU should be priced lower than the initial processor complex.

Semiconductor Memory and Logic

"Conservative" IBM put the stamp of approval on semiconductor memory with the introduction of System/370 MOSFET storage technology in August, 1972.

Various types of memories which use bipolar and MOS technologies are currently available. Random access memories (RAM), read-only memories (ROM), programmable ROMs (PROMs) and electrically-alterable ROMs (EAROMs) are all part of the current product offerings from the independent semiconductor industry.

The RAM is used for main memory and is presently available in both bipolar and MOS forms. MOS currently enjoys substantial advantages over bipolar because of its higher packing density, its improved access speeds, lower manufacturing complexity and lower cost, and should become the dominant technology by the end of this decade.

Intel has been enormously successful in its pioneering efforts in the development of a 1K, p-channel, MOS, dynamic RAM (dynamic RAMs need periodic refreshment of the signal levels). Access time is currently 300 ns, cycle time is 500 ns and power dissipated is 100 microwatts per bit. The 1103 was initially introduced in 1970 at a price of $60. By 1973, the price was averaging $4, or .4 cent per bit.

Current efforts in the semiconductor industry are being directed toward the achievement of production quantities of 4K and 16K, n-channel, MOS, dynamic RAMs. (IBM has already indicated that 32K MOS RAMs are available in the laboratory.) The n-channel product is highly desirable because of higher speed characteristics compared to p-channel. In addition, chip size will increase and rapid price reductions may be anticipated once the manufacturing process has stabilized.

MOSFET devices have found wide use in the integrated circuit industry because they are naturally suited to planar surfaces. Thus they are ideal for production of large arrays by standard processing of silicon wafers. Field effect transistors (FETs) occupy only a fraction of the chip area required for bipolar.

In addition, the manufacturing process to produce an MOS integrated circuit requires fewer steps than for bipolar, so that significant

cost reductions may be achieved. The price-per-function advantage for MOS over bipolar frequently is as high as 10 to 1. (Note that IBM recently switched from bipolar to MOS chips for the 370/138 and 148.)

The graph in Figure 11-2 shows cost per component for bipolar and MOS technologies. After 1970, MOS dominated the pricing structure. In that year, the cost per component was one cent with 1,000 components per IC. Memory (RAM) prices declined by about 30% to 40% annually between 1971-77 and the extrapolated values indicate a cost per component by 1980 approaching .001 cent, with a half million to a million components or bits per IC.

The chart in Figure 11-3 shows the silicon family tree and the various devices and circuit types.

Over the years, the ability to manufacture core has improved to the point where prices are about .2 cent per bit. Typical core speeds are concentrated in the 500 nanosecond to one microsecond access time range. Now, however, high-speed semiconductor memory has become economically available in both bipolar and MOS forms, and mid-1975 was probably the turning point. Intel was selling its 4K Model 2107A-8 (a 420 ns MOS device) for less than $12 in lots of 100 and it was anticipating .1 cent per bit to be a realistic goal in 1977.

The speed requirements of mainframe memory can be met by bipolar and silicon-on-sapphire (SOS) devices. While core, plated wire and MOS memories have in the past satisfied some of the speed requirements for large-scale usage, only bipolar and SOS devices appear to qualify at speeds of less than 40 nanoseconds.

The SOS manufacturing technique requires that very thin layers of silicon be deposited on a flat substrate of synthetic sapphire. Parasitic capacitance is reduced by a factor of 20 compared to standard MOS transistors. When MOS circuits are manufactured with this process, they combine the high speed attainable by bipolar transistors with the packing density of MOS. At present, SOS structures are more expensive to fabricate than MOS.

Forecasts call for NMOS, bubbles and charge-coupled devices (CCD) products to equal or exceed the price-per-bit of core after 1976. By 1977-78, bubbles and CCDs could be at prices below .1 cent per bit. PMOS, which is already available at prices approaching those of core, has speed disadvantages but can perform many main memory tasks. Slower speed MOS memory chips are combined with high-speed bipolar caches (buffers) in System/370 Models 158 and 168, Amdahl 470V/6 and V/7, the Univac 1100/80 line, Burroughs 7800, Honeywell Level 66, etc.

Logic

While general purpose mainframes have declined in price over time, typical minicomputer prices (with 32K bytes) have declined more rapidly, i.e., 18%-20% annually between 1971-77 (e.g., HP-2100A in 1967 versus the HP-21MX in 1977). These significant improvements are directly related to IC developments. Today, only about 10% of a typical minicomputer's value is composed of integrated circuits.

Logic families are closely related to IC development. In the late '50s, resistor-transistor logic (RTL) seemed to be the best way to go. By 1962, diode-transistor logic (DTL) had appeared. Transistor-transistor logic (TTL) was announced in the early 1960s, while initial versions of emitter-coupled logic (ECL) were introduced by Motorola in 1962. (IBM used versions of ECL in the 7030 Stretch Machine in the mid-'50s.)

The basic requirements for any logic family are:

- Flexibility.
- Speed.
- Availability of complex functions.
- High noise immunity.
- Wide operating temperature range.
- Low power dissipation.
- Lack of internally generated noise.
- Low cost.

TTL (in various forms) and ECL are the primary surviving bipolar logic families used by most computer designers today. IBM has utilized a current switched logic (ECL) for its System/370 computers. While such logic has the advantage of higher speeds, it is also generally true that as speed increases so does power dissipation. Removal of circuit-generated heat is essential for reliable operation.

Figure 11-4 compares logic families in terms of speed and power dissipation. Schottky TTL is a high-speed version of the standard TTL. From the table it can be seen that the progression from RTL to ECL leads to increased speeds and higher dissipated power. Circuit designers currently have several choices available for optimization of speed and dissipated power.

Integrated-injection logic (IIL) is a new design which will increase bipolar circuit density. These devices could move out of the laboratory

next year. At IBM, in Boeblingen, Germany, the terminology used for the same basic approach is merged-transistor logic (MTL)*.

The IIL approach utilizes gates which do not require current source resistors. The space required for an entire gate is 1 to 2 square mils, or the space needed for one transistor in the TTL family. It is capable of propagation delays approaching 5 nanoseconds with power dissipation as low as one *nanowatt* per gate.

The above advancements in ICs (ROMs, RAMs, PROMs, EAROMs, etc.) will permit "plug-in firmware" to be practical by the end of the decade. Burroughs already uses this concept in the Burroughs Scientific Processor (BSP), and TI's calculators (Models 58/59) use it today.

Main Memory (RAM)

The IBM 370/158 and 168 main memory requires nine times less area per cell than the older 360/85 and recent charge-coupled devices (from TI) of the one-cell variety have substantially increased relative bit density even further. However, the technology is quickly approaching a point where it will not pay to stretch it any further, using *current* photolithography methods; i.e., the potential advantages from increased density versus the technical problems of cooling, achieving the incredibly small line widths and so forth will yield less of a payoff in the future.

In terms of photolithography, currently 6 microns is typical and this can be reduced to 2 microns using shorter wavelengths or electronic beam exposure techniques. IBM's near-term objectives appear to be to achieve 500 to 600 circuits on a one-quarter inch by one inch wafer.

The IC Leaders and Recent Trends

Total worldwide consumption of all types of ICs (not just memory devices) has climbed to nearly $4 billion in 1977. Figure 11-5 shows the memory circuit market share breakdown.

Top-tier companies include: TI, Motorola, Fairchild and IBM (internal supplier only). Second-tier firms are: Mostek, National Semiconductor, Intel, Advanced Micro Devices, RCA, ITT and Zilog, among others. Recent high-speed, static and dynamic RAM competitive products (components) are shown in Figure 11-6.

---

*Merged-transistor logic has the density of MOS with the speed of bipolar; this chip may be ready for production in the 1978-79 period.

IC Storage in Japan

On June 7th, 1976, Fujitsu announced that it had succeeded in developing a 64K-bit RAM chip. The characteristics include 64,000 bits on silicon substrate, performance in the 100- to 200-nanosecond access time range and substitution for electron beam and x-ray exposure techniques by current semiconductor fabrication processes.

Although this product has been developed in the lab, we should not expect 64K chips to be available in volume quantities before 1979.

This Japanese development was reported by Japan's Ministry of International Trade and Industry (or MITI). The MITI project is comprised of five mainframe computer makers which were recently restructured into two groups: the Fujitsu/Hitachi/Mitsubishi and the NEAC/Toshiba groups. The MITI project has a budget of approximately $400 million, and will concentrate on very large scale integration (VLSI) logic and memories and other circuits. The MITI total budget over the 1977-80 period should approach $100 million for this VLSI activity alone, while an additional $190 million will be spent on a "fifth generation" computer and about $117 million will be allocated to pattern information processing R&D.

Optic Communication Links

The significant advantages of fiber-optics in computers include wide bandwidth in the microwave range, no electronic interference, electronic isolation and loss of only 2 DB per kilometer in glass fibers now in use (which represents far better integrity than that characterizing microwave techniques).

Some of the work to be done in fiber-optics includes the development of a reliable laser source, fast detectors and new coupling techniques.

The current industry leaders of the above "new" technologies are:

- Electron-beam Lithography: Bell Labs, IBM, Texas Instruments and the Japanese companies.
- Bubble Memories: Bell Labs, IBM, Rockwell, Texas Instruments and the Japanese companies.
- Fiber-Optics: Bell Labs, Corning Glass, ITT, IBM and the Japanese companies (Fujitsu, Hitachi, NEC, etc.).

Subsystem Technology Forecast

Diskettes will have increased capacity through a combination of greater density and the ability to record on both sides. These improvements will increase effective capacity more than fourfold to over one million characters.

Current and future terminals (IBM "Orbit/Trinity") and other small systems will use the floppy disk. In addition, stacks of floppies (hundreds) may be used as a low-cost, high-capacity alternative for archival storage.

Medium Capacity Disks

"Gulliver" is IBM's code name for the new fixed disk which is currently integrated within the 3790, 3600 and other terminals. We can expect this important device, with a current capacity range of from five million to over 27 million characters, to play a major role together with removable diskettes in future small business systems, minicomputers and other intelligent terminal products. The recording density, speed and other attributes of the announced "Gulliver" file are unknown at this time. A 50- to 100-megacharacter "Gulliver" file is anticipated before 1980, however.

Large Capacity Fixed Disks

The original IBM code name for this nonremovable device was "Apollo." It is to utilize newly improved multilevel recording techniques, increasing capacity to 11,000 bits-per-inch and about 800 tracks-per-inch. Such a device would accommodate in excess of one billion characters per drive. IC multi-element heads would be used (Burroughs has already announced its own, new IC-head device), and average positioning time would probably be in the 10-15 ms range. This disk, together with the IBM 3350 and improved versions of the 3344 (more read/write heads), should be available on both System/370 and System/80. Announcement of the "Apollo" is anticipated for early 1978, with volume deliveries to follow in 1979.

Disk improvements in terms of capacity/$ were significant in 1965 (threefold improvement) and have been accelerating again during the past five years (1971-76) (see Figure 11-7).

Bubble Memories

Texas Instruments' Silent 700 terminal uses 92K bit bubble chips and Hitachi has a POS terminal which also uses bubbles for storage. IBM has built a device incorporating one micron diameter bubbles with access times ranging from 1 millisecond to as low as 100 microseconds depending on the configurations and loop lengths. Bubble size appears to be material-dependent. The packing density is dependent on the line size that can be laid down on the bubble material, and is also dependent on the exigencies of photolithographical technique.

In contrast to charge-coupled devices (CCD), the bubble process is more complicated and, according to IBM, bubbles are being emphasized more for storage technology, whereas CCDs are being directed towards analog device uses such as scanner applications. The economic crossover point for some additional applications for bubbles will occur in 1978 (0.1 cent per bit, 1 megabit density) and 1980 (0.03 cent per bit, 3 megabit density). During that period, fixed-head disks and some diskettes will be replaced by bubble devices.

Non-Impact Printers

A magnetic ink-jet printer has recently been introduced by IBM for the office equipment or word processing business of OPD. This device has a print speed of up to 92 characters per second, but is relatively expensive. More ink-jet printers of higher speed are expected to be introduced over the next five years. Devices with speeds exceeding 200 characters per second which incorporate multi-element ink-jet heads can be anticipated. Advantages of ink-jet devices are better resolution, enhanced print quality, lack of noise and potentially higher reliability. High-speed line printer price/performance trends are shown in Figure 11-8.

Mass Storage Devices

Additional versions of the IBM 3850 are anticipated; a larger model with twice the current density (or 940 billion characters), and a smaller version for the new 370/138 and 148 systems are expected to be introduced by 1978.

Within the IBM System/80 timeframe, a 4X capacity model should be available and other advanced structural improvements can be forecasted; e.g., using bubbles or CCDs as another level in the storage

hierarchy; faster staging disks (more read/write heads); and, most important, a cache-type buffer (similar to the 158 or 168) which could significantly enhance data availability and overall system performance by improving I/O command queuing and data access time across many staging devices. Mass storage devices will soon obsolete magnetic tape drives which are at the trailing edge of their price/capacity (Figure 11-9).

The chart in Figure 11-10 attempts to put all the various storage technologies together so that one can appreciate the distinct spectra of capacity and cost and the relative improvements over time (past, present and future).

1. MOSFETs will dominate and cover most on-line, high-speed main storage needs during the next ten years.
2. Bubbles should replace most fixed-head disks, drums and diskettes by 1980.
3. Moving-head disks will continue to improve in capacity and cost and no replacement device is anticipated for at least ten years.
4. Mass storage devices (e.g., the IBM 3850) of various designs are quickly closing the "gap" between on-line disk/tape and off-line disk.
5. Archival files are anticipated by 1983-85 as a cost-effective replacement for off-line magnetic tapes and disks in terms of easier access, greater speed and equivalent price per bit.

Gas Panel Displays

Some plasma displays have already been introduced by both Burroughs and IBM, in particular the IBM 3600 financial terminal. Additional displays of this type are expected over the next five years. IBM plans to build a 14" x 14" panel with multilevel addressing techniques to reduce the cost of drivers.

Advantages of plasma displays are high resolution, flicker-free operation, greater reliability, inherent storage (i.e., no requirement to refresh the display screen) and multicolor capability. The flat design permits them to be used in many unique arrangements for terminals or consoles.

Josephson Technology and the Larger View

A thorough review of recent IBM technical publications and System

Journals, and a tour of the T. J. Watson Research Center suggests that the successor to System/80 *can* emerge by 1985-87 (if IBM decides to announce it in that period) and that the successor mainframes (CPU and memory) will be far more revolutionary with respect to System/80 than System/80 will have been relative to System/370.

The technological breakthrough underlying this phenomenon is called "Josephson Tunnel Junctions"* (named after Dr. Brian Josephson, who discovered the Josephson effect in 1962).

Josephson devices (logic and memory) are superconductors which operate at the temperature of liquid helium (4.2° Kelvin), with switching speeds in the picosecond range (100 to 200 times faster than today's fastest semiconductors). They are of three-layer, laminated construction with outer layers of superconducting materials separated by an insulating layer, and they generate only one-thousandth as much heat as do current techniques.

The entire Josephson computer, including memory (but excluding I/O devices), may be encased in liquid helium with an outer chamber of liquid nitrogen. Recent technological fallout from the NASA program has solved most of the cryogenic problems. IBM has been actively working on Josephson devices for over six years.

The key questions are *not* technical (although some problems remain to be solved), but rather:

- Whether IBM corporate management will decide to impact System/80 in the 1985-87 timeframe.
- Whether it makes good business sense to offer this kind of horsepower (i.e., whether the marketplace can absorb it or whether a 100- to 200-fold performance increase will saturate the marketplace).
- Whether future peripherals can keep up with superconductor technology (or how "balanced" systems can be created).
- What the competitive environment will be like during that period.

Josephson capability is *not* limited to large-scale, high-priced processors — the technology can cover a very wide spectrum of systems—it is, in fact, a very small (physically speaking), large-scale computer.

---

*See IBM patent #3,849,276 November 1974.

A Josephson multiplier was described recently in the *IEEE Journal of Solid-State Circuits* and in the May, 1976 issue of IBM's *News/ Research.* The multiplier, the most complex Josephson circuit yet presented in the literature, contains 45 gates (a gate performs a single logical operation such as "and" or "or"). Projections indicate that two four-bit numbers could be multiplied in 12 nanoseconds.

The product of power dissipation and delay per logical operation, a standard figure of merit in switching circuits, is about *1000 times better* in the Josephson multiplier than in present semiconductor circuitry. Furthermore, detailed test results reported in the *IEEE Journal* indicate that "the various design considerations—signal levels, timing, impedances—are sufficiently well understood to provide high performance and reasonable tolerances. The tests also confirmed the substantial progress that has been made in fabrication techniques for sizable arrays of Josephson devices and supporting circuitry."

According to Dr. Eugene Fubini, a private consultant on science and technology (former IBM Vice President and Group Executive responsible for Research and Advanced Systems Development from 1965 to 1969), "in seven to eight years (1983-84) we should expect 10,000 times greater performance [from Josephson technology] . . . a megabit storage chip will cost approximately $30 by 1985."

Advanced Computer Techniques, Inc. believes that 10,000 times greater performance is *not* probable at that time; however, a 100- to 500-fold improvement *is* practical by 1985, and this is sufficient to change our "style" of programming.

These incredible speeds and capacities will dramatically change the way we program future computers. There will no longer be much concern about efficient code or wasteful use of main memory.

In fact, programmers/analysts can try *all* possible solutions to a problem since it will take a few nanoseconds per try. Trial and error, interactive programming and analysis would increase greatly with this kind of capability, and drastically change the way we create software. Dr. Fubini also states that while "in the past we were problem rich and solution poor," Josephson technology will reverse the situation, making us "solution rich and problem poor" in the near future.

The initial markets for Josephson devices and computers will likely arise in connection with federal governmental projects, military applications and weather forecasting, and eventually spill over into the commercial section by 1985-87. A key question is how can we utilize (program) this technology *effectively* when one second of computer time make take a thousand man-years to program.

I believe that any advanced system planned for shipment in the 1979-80 timeframe must reflect an awareness on the part of the vendors of the impact of Josephson technology in 1985-87. Will the hardware and software architecture of the 1980s be adaptable to Josephson technology? Will the major manufacturers invest sufficiently in this revolutionary technology to remain competitive in the 1980s? What new programming techniques will be necessary to tap this incredible resource?

Computers are today just beginning to affect our lifestyle, and we have probably reached the point where we cannot return to the manual information handling process in many application areas, e.g., in airlines reservations, process control, command and control, etc.

While it may yet be possible to argue that if you destroyed all the computers in existence today it wouldn't *seriously* affect your life, it seems equally obvious that by the early 1980s we would be drowning in paperwork. Progress would necessarily retreat without the powerful computer systems and networks upon which a service-oriented society is—we would say inescapably—dependent.

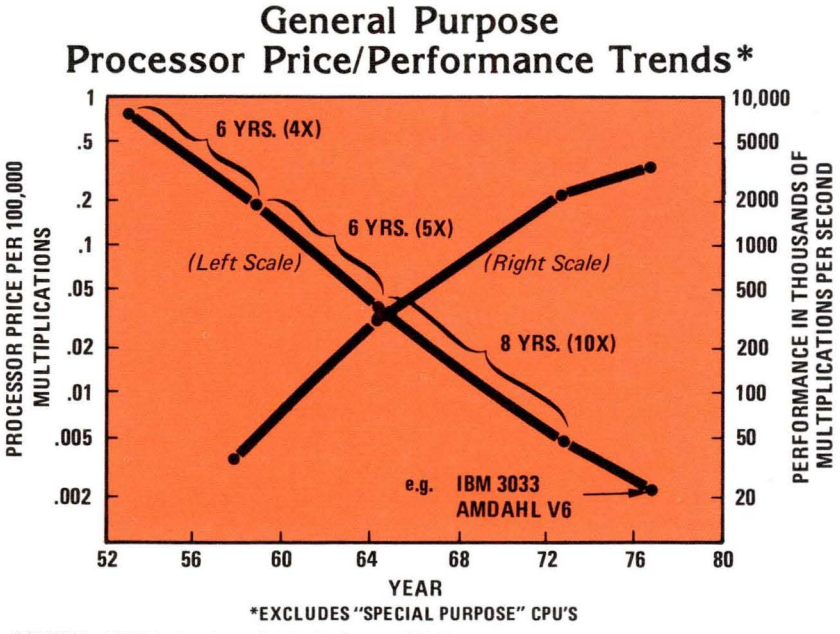
Nor is this an alarming picture; computer technology can directly help to solve the energy crisis and pending natural resource crisis. Some examples: paper will become the *exception* medium for human communication, being replaced by on-line storage and retrieval devices; microprocessors will improve transportation efficiency, reducing fuel consumption; there will be less need to travel when low-cost computers and networks permit effective human interaction from a terminal in the office or home.

And products—appliances, automobiles and other machines—can be designed to last longer (through microprocessor failure prediction/detection and self-repair/adjustment), thus reducing, or at least holding the line on, the amount of energy and natural resources required to manufacture a product. The current "planned obsolescence" game will itself become obsolete in the face of resource shortages and the resulting demand for products with built-in "survival" factors of two to five times current product life expectancy.

Of course, the fact that we *can* do something doesn't mean we have the will to do it. However, the expected shortages of some of the world's prime resources will generate an almost irresistible demand for more intelligent, *purposive* use of such of those precious resources as remain.

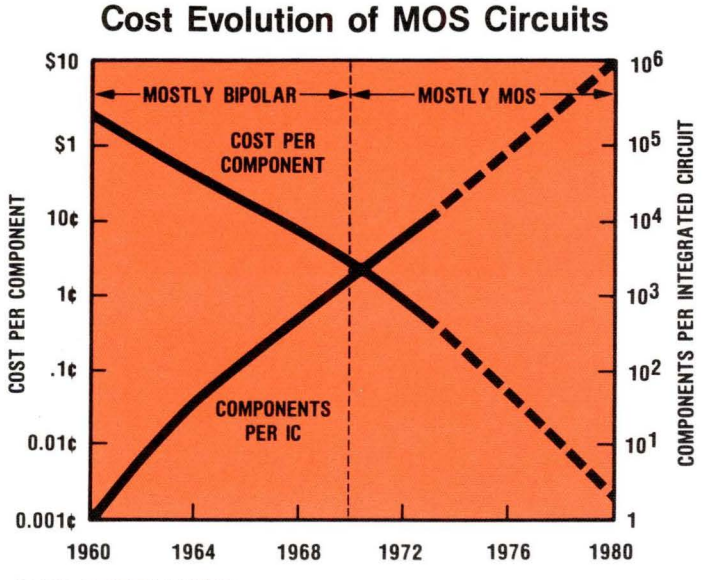
Thus, on one level, the demand for computer power, storage and easier access to vital data will ensure a new generation of dramatic growth in the computer industry (however altered in shape and character

the industry will be by the mid-1980s). But most important of all, this growth will be one of the fundamental factors in ensuring the survival of the community of man, in a way that makes the prospect of survival less grim than it otherwise might be. The computer industry can and must be in the forefront of a movement away from artificial services and towards necessary, truly life-enhancing ones. It will be good—and gratifying—to participate in an "industry trend" with such potential.



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-1

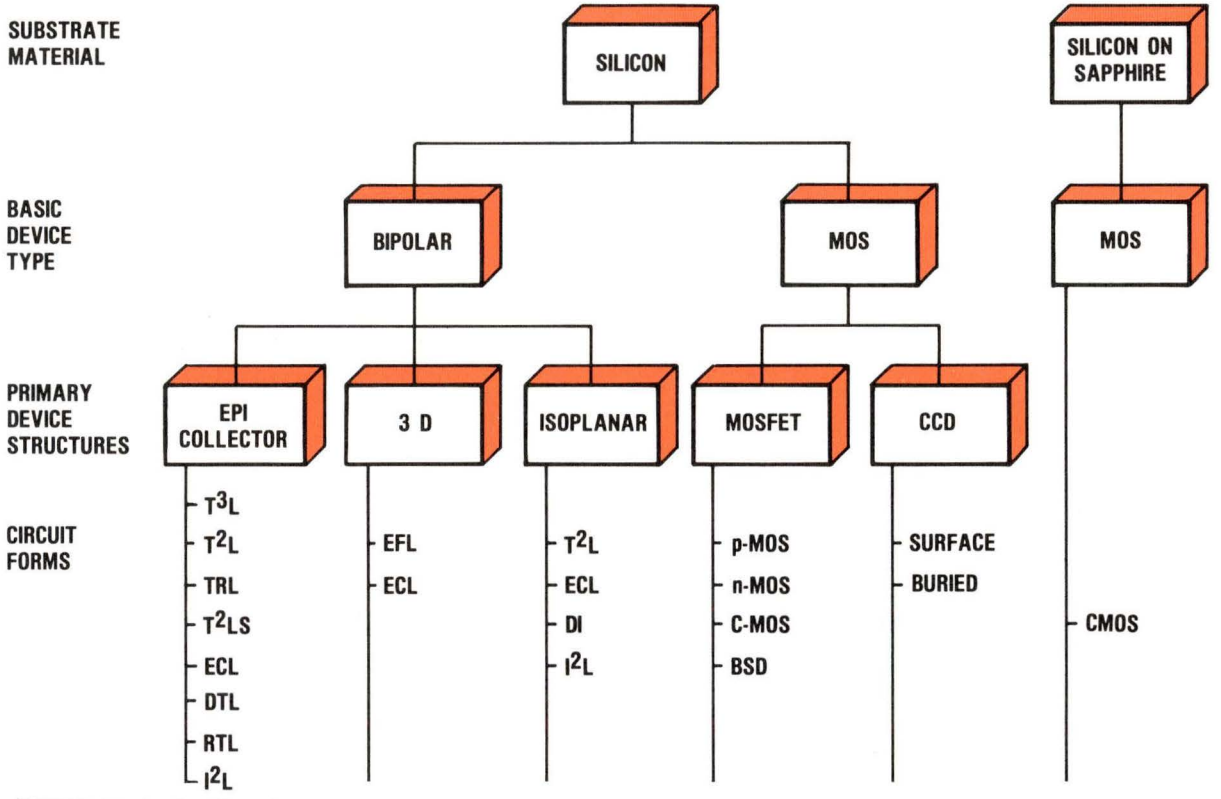


SOURCE: SCIENTIFIC AMERICAN

SOURCE: Scientific American

Figure 11-2

Silicon Technology



SOURCE: Electronics Magazine

Figure 11-3

Bipolar Logic Family Comparisons

FAMILY	SPEED PROPAGATION DELAY PER GATE ns	POWER DISSIPATION PER GATE mW
RTL	12	12
DTL	30	10
TTL (STANDARD)	10	10
TTL (SCHOTTKY)	3	20
*TTL (LOW POWER SCHOTTKY)	5	2
ECL	1-2	55-100

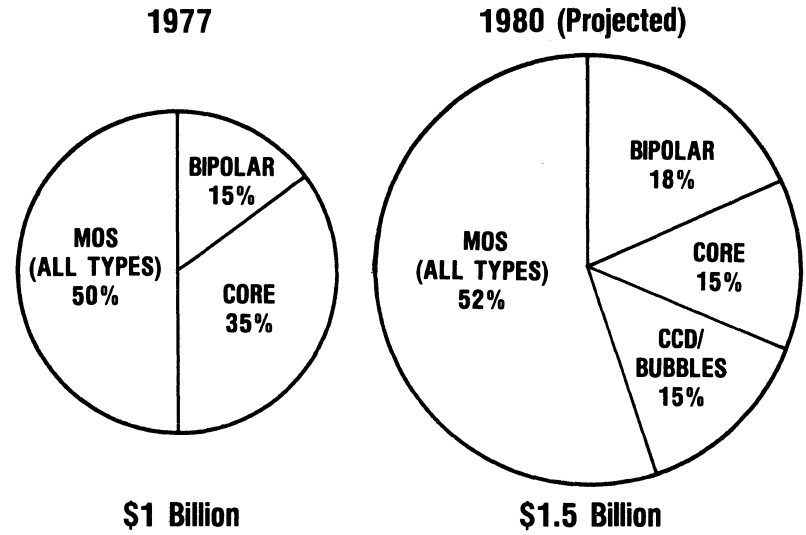
NOTE: ns — nanosecond
mW— milliwatt

*LOW POWER SCHOTTKY IS THE LATEST HIGH GROWTH COMPONENT.

SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-4

Worldwide Memory Market Trends



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-5

Static RAMS

MANUFACTURERS	MODEL	TYPE/DENSITY	SPEED	POWER
FAIRCHILD	93415	BIPOLAR/1K	35ns	1 WATT
INTEL	2115A	MOS/1K	45-70ns	3/4 WATT
INTEL	2115AL	MOS/1K	45-70ns	1/2 WATT
INTEL	2147	MOS/4K	55ns	1/2 WATT

Dynamic RAMS

MANUFACTURERS	MODEL	TYPE/DENSITY
TI	TMS4071	MOS/16K
MOSTEK	4116	MOS/16K
MOSTEK	4117	MOS/16K
NEC (Japan)	4117 (equiv.)	MOS/16K

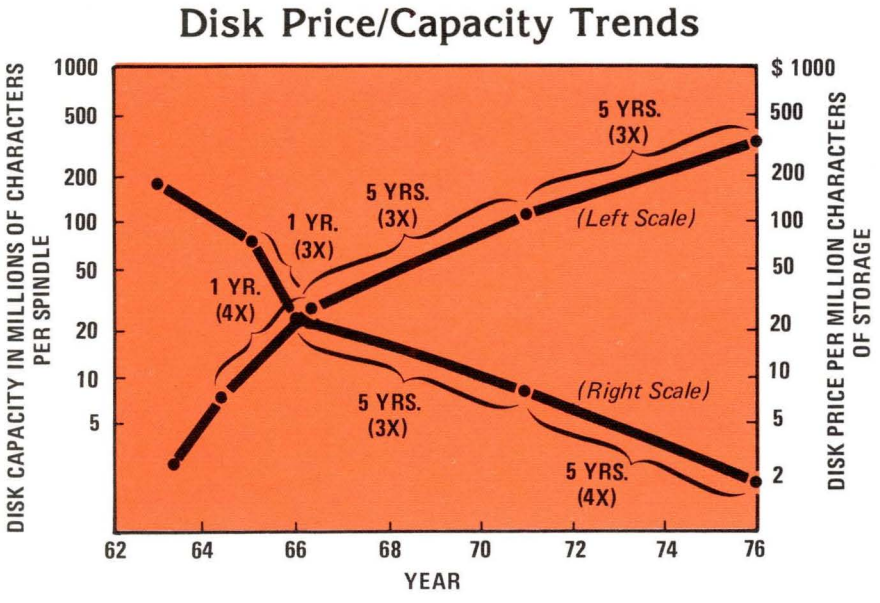
EAROMS

MANUFACTURERS	MODEL	TYPE/DENSITY
INTEL	2708	MOS/8K
INTEL	2758	MOS/8K
INTEL	2716	MOS/16K
TI	2716	MOS/16K

*NOTE: MERGED-TRANSISTOR LOGIC HAS THE DENSITY OF MOS WITH THE SPEED OF BIPOLAR AND THIS CHIP MAY BE READY FOR PRODUCTION IN THE 1978-79 PERIOD.

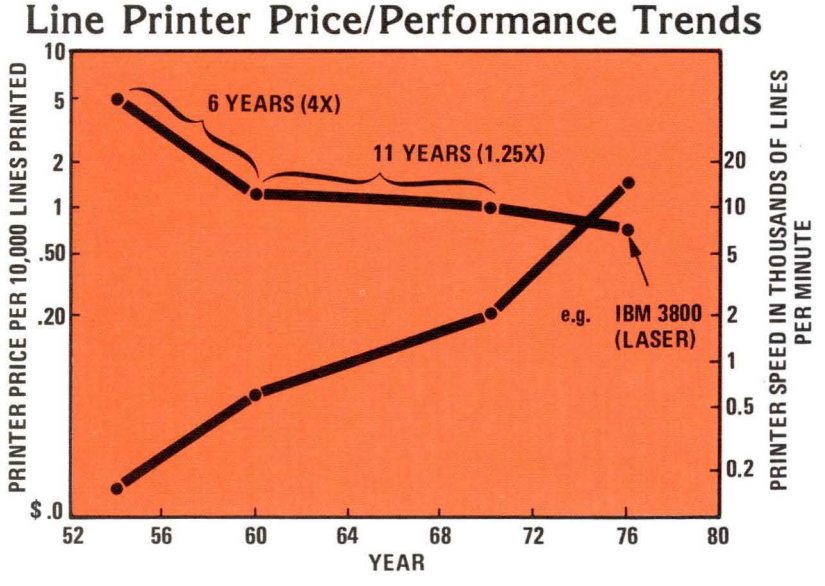
SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-6



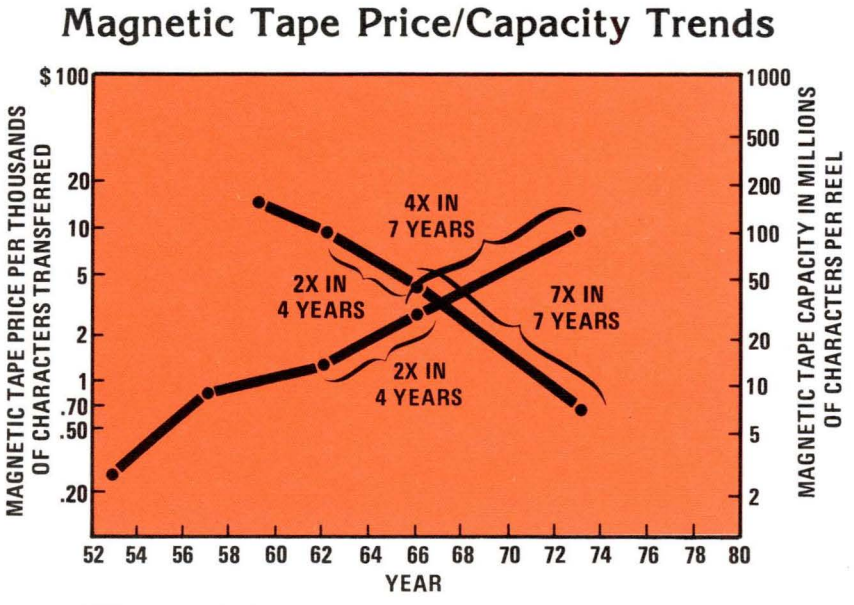
SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-7



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-8



SOURCE: ACT/Technology-Analysis Group (TAG)

Figure 11-9

NOTES

ISBN: 0-931336-00-7